AN EVALUATION OF SOFTWARE DEVELOPMENT PRACTICE AND ASSESSMENT-BASED PROCESS IMPROVEMENT IN SMALL SOFTWARE DEVELOPMENT FIRMS

Submitted by

Aileen Patricia Cater-Steel B.Bus (USQ), MIT (USQ)

School of Computing and Information Technology

Faculty of Engineering and Information Technology

Griffith University

Submitted in fulfilment of the requirements of the degree of

Doctor of Philosophy

December 2004

Abstract

As software becomes increasingly important to all aspects of industry, there is a need to encourage practitioners to adopt best practice so as to improve the quality of the processes in use, and therefore achieve targets relating to time, budget and quality. The software development industry in Australia is dominated by a myriad of small firms. This presents a challenge in terms of determining the current practices of industry participants, and in devising improvement initiatives which are feasible for small organisations. Currently, the level of adoption of best practice among local software developers is unknown. To help improve the software industry, it is necessary to determine the current status of use of practices and techniques. Furthermore, the effectiveness of assessment-based software process improvement for small organisations needs to be evaluated. The objective of this research is to understand the extent of software development practices currently in use, and to evaluate the effectiveness of assessment-based software process improvement initiatives for small firms.

To achieve this objective, an extensive mail survey of the Queensland software industry was conducted to identify and compare *best practice* in software development with *current practice*. The survey was based on the software best practice questionnaire used by the European Software Institute. Following on from this, a detailed evaluation of a process improvement program in 22 small firms was carried out. The program used the Rapid Assessments for Process Improvement for software Development (RAPID) model and method. RAPID is based on ISO/IEC 15504 (SPICE) and includes eight processes: requirements elicitation, software development, configuration management, quality assurance, project management, problem resolution, risk management, and process establishment. The evaluation analysed the process capability of the firms as reported from one-day software process assessments and also the extent of improvement as recorded at follow-up meetings held 7 to 16 months after the assessment. Both quantitative and qualitative techniques were used to analyse the assessment reports.

The study confirmed that there is wide variation in the extent of adoption of software development best practice in terms of the individual practices, as well as the organisations. While project management planning and customer involvement practices are widely adopted, the use of metrics for estimating and testing are barely used by the organisations that responded to the survey. Overall, practices of a technical nature are more widely adopted compared to techniques related to support and management. Organisations involved in developing commercial off-the-shelf software have higher adoption than firms which do not develop such systems, and adoption of best practice is associated with the size of the development group. The leaders in adoption have significantly better practices when compared to the laggards for 40 of the 44 practices included in the survey. Furthermore, organisations from the finance, insurance and utilities sectors exhibited higher adoption of best practice compared to organisations from other sectors. The overall adoption of 48 percent implies that the organisations which responded have adopted, on average, almost half of the best practices in the questionnaire. While this overall adoption rate places the Queensland software industry in a competitive position compared to adoption of firms in European countries, there is scope for improvement.

The process improvement assessments of 22 firms also confirmed that the capability of technical processes is higher than that of management processes; and suggested that higher capability is associated with the proportion of experienced staff and the proportion of staff with post graduate qualifications. Higher process capability is also associated with firms undertaking projects of lengthy durations. Most of the processes were rated at the lowest levels. Almost one third of all the processes were rated as *incomplete* (level 0) and 46 percent were rated as *performed* (level 1).

The evaluation of the process improvement program was conducted by analysing the 22 assessment reports, and the 20 final reports from the follow-up meetings. The extent of improvement is associated with the proportion of technical staff and the proportion of formally qualified staff. The evaluation revealed that assessment-based process improvement programs are effective for small firms, regardless of the maturity of the processes at the time of the assessment.

As well as detailing the process capability of 22 small software firms, this study provides an interesting insight into the actions, reasons for inaction, and reactions of the firms as far as implementing the recommendations from the assessments. Analysis of the reactions of the participants of this program suggests that for small firms, mentoring, training and organisation stability are important factors, while senior management support may not be an issue of concern.

The study indicates that small firms can benefit from a low cost process improvement program with a restricted scope, a short time frame to evaluation, and mentoring from external assessors/consultants. It is also crucial that the firm is not disrupted by internal or external events during the course of the software process improvement program. Furthermore, this study provides a contribution to assessment methods by validating the RAPID model and method, and providing recommendations to improve the RAPID method. The outcomes from this research have the potential to better equip practitioners and consultants to undertake software process improvement, hence increasing the success of small software development firms in domestic and global markets.

Certification of Thesis

I certify that the ideas, analyses, results, and conclusions reported in this thesis are entirely my own effort, except where otherwise acknowledged. This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

During the course of the research, a number of research papers were published. For all of these works, I was the sole author:

1999 'SEA National Industry Improvement Program "Best Practice" Survey: Preliminary Findings', paper presented to Software Engineering Australia '99, Canberra.

2000 'Software process evaluation: experience report from Down Under', in *Proceedings of European Conference on IT Evaluation*, Dublin, pp. 353-63.

2000 'COTS developers lead best practice adoption', in *Proceedings of Australian Software Engineering Conference (ASWEC)*, IEEE, Canberra, pp. 23-30.

2001 'Process improvement in four small software companies', in *Proceedings of Australian Software Engineering Conference (ASWEC)*, Canberra, pp. 262-72.

2002 'Process capability assessments in small development firms', in Proceedings of IASTED 6th International Conference Software Engineering and Applications, Cambridge, MA, pp. 737-42.

2002 'Software process evaluation: experience report', *Electronic Journal of Information Systems Evaluation*, vol. 5, no. 2.

2003 'SPICE-based software process improvement: an empirical study of 22 small Australian firms', paper presented to Australian Software and Systems Engineering Process Group Conference, Surfers Paradise, 24-26 September.

2004 'Low-rigour, rapid software process assessments for small software development firms', in *Proceedings of Australian Software Engineering Conference (ASWEC)*, Melbourne, pp. 368-77.

In addition to the papers listed above, one paper was co-authored with my supervisors Terry Rout and Mark Toleman: my role was that of primary author:

2004 'After the assessment: actions and reactions of 22 small Australian firms', in *Proceedings of 4th International SPICE Conference on Process Assessment and Improvement*, Lisbon, Portugal, pp. 54-63.

Signature of Candidate

Date

Acknowledgements

This thesis is the result of almost seven years of part-time study, and during this time I have been accompanied and supported by many people. I am delighted to now have the opportunity to express my sincere gratitude and appreciation to all of them.

I am deeply indebted to my principal supervisor, Associate Professor Terry Rout for his stimulating suggestions, guiding hand and expertise in the field of software process improvement. My associate supervisor, Associate Professor Mark Toleman (USQ) has given much of his valuable time to help me over the last two years. In particular, Mark’s encouragement, optimism and research expertise has enabled me to overcome many obstacles. I recognise, with gratitude, the support from Software Engineering Australia (Queensland), and the Software Quality Institute at Griffith University.

My colleagues at USQ have provided support, interest and advice, in particular, Carol Edwards, Chris O’Reilly, Ilona Eberle and members of the Department of Information Systems. Special thanks also to my dear friends Rosie, Edel, Jill, and Annie, and my brothers and sisters for their continued encouragement and affection.

Finally, my thanks go to three very special people. My two children, Emily and Declan have helped me achieve my goal through their patience, understanding and willingness to accommodate my dream. I am also deeply grateful to my husband, Michael for his love, patience and encouragement.

> *Persistence*
> *‘Nothing in the world can take the place of persistence.*
> *Talent will not; nothing is more common than unsuccessful men with talent.*
> *Genius will not; unrewarded genius is almost a proverb.*
> *Education will not; the world is full of educated derelicts.*
> *Persistence and determination alone are omnipotent.*
> *The slogan ‘Press on’ has solved and always will solve the problems of the human race’* (Calvin Coolidge, President USA 1923-1929).

Table of Contents

Table of Figures

Table of Tables

Appendices

1 CHAPTER ONE—INTRODUCTION

1.1 Research topic

This thesis reports on a comprehensive attempt to determine the extent to which software developers in Queensland are using best practice, and to empirically assess the effectiveness of a process improvement intervention in small software development firms. The objective is to understand the extent of software development practices currently in use, and to evaluate the effectiveness of assessment-based software process improvement (SPI) initiatives for small firms. To achieve this objective, an extensive survey of the software industry was conducted to identify and compare *best practice* in software development with *current practice*. Following on from this, a detailed evaluation of a software process improvement initiative in 22 small firms was carried out.

This chapter provides the foundation for the thesis. Firstly, background information describing the Queensland software development industry is provided and the significance of the research is established. Next, the research problem is stated and the research is justified on theoretical and practical grounds. The format of the thesis is then outlined and definitions are provided. The final section describes the scope of the research, justifies the boundaries and states key assumptions.

1.2 Background to the research

Prior to providing background information about the local software industry, the importance of software and software improvement is discussed.

1.2.1 Importance of software process improvement

For most organisations, and increasingly for individuals, software has become an essential element for survival. As well as in business systems, software is a key component of telecommunications, defence, transport, and medical systems. Software also plays a vital operational and strategic role enabling organisations to meet challenges of flexibility and time-to-market, and to reduce costs and maintain quality (Dromey & Rout 1992; Geck et al. 1998). With the growing use and ubiquity of internet and mobile technologies, and embedded software in consumer products, individuals are also more reliant on software; it has become ‘woven into the threads of our daily lives' (Glass 1996, p.11). Today, software is an intrinsic part of cars, watches, televisions and many other commodities used every day (McKerlie Consulting 1996).

Due to the growing dependence on software, problems in developing software can have devastating results at all levels: individual; business; community; national and international. Consequently, the improvement of processes associated with software development has become a focus for practitioners and researchers alike (Dyba 2000; Hall, Rainer & Baddoo 2002; Niazi, Wilson & Zowghi 2003).

In 1987, Fred Brooks observed that 'the gap between the best software engineering practice and the average practice is very wide—perhaps wider than in any other engineering discipline' (Brooks 1987, p.15). The view that the *state of practice* lags far behind the *state of the art* is still held today (Ludewig 2001; McConnell 2002).

Recently, software engineering research has been criticised on two counts. Firstly, that researchers, while eagerly prescribing practices, are not aware of what practitioners are actually doing (Fitzgerald 1997, p.202; Glass 2003b); and secondly, that many of the prescribed practices, methods, techniques and standards are not thoroughly evaluated (Fenton, Pfleeger & Glass 1994; McBride 2004) and may not even be suitable for all software development organisations (Brodman & Johnson 1997a; Bucci, Campanai & Cignoni 2001; Kautz 1998b; Pfleeger et al. 1997; Richardson & Ryan 2001; Wilkie, McFall & McCaffery 2004).

1.2.2 Software industry

The importance of the software industry in Australia was recognised with the founding and ongoing support of Software Engineering Australia Limited (SEA). SEA has been operational since 1999 as a not-for-profit association, funded under grants and in-kind contributions from the Australian federal government, state governments, universities and the private sector. SEA's initial business plan stated its aim as the coordination of national expertise and resources to deliver internationally competitive software engineering skills throughout Australia (SEAQ 1997). Each state-based resource centre offers a range of facilities, including information services, education and training, technical problem solving and process improvement activities.

Although comprehensive and consistent data on the Australian software sector is lacking (Framework for the Future Mapping Working Group 2002, p. 8), a brief overview of the Australian software industry is provided, followed by a profile of the software industry in Queensland.

It is estimated that globally, US$182 billion was spent on software in 2001, with Australian spending of software accounting for 1.3 percent of the global spend (McKinsey & Co. 2002, p. 23). The Australian software industry makes a significant contribution to the Australian economy by creating jobs, and increasing productivity, capability and competitiveness (BSA & PricewaterhouseCoopers 1998). In 2001, the software and services sector generated revenues of A$2.6 billion (Framework for the Future Mapping Working Group 2002). In Australia, the Information, Communications and Technology (ICT) industry is dominated by small firms: almost 96 percent of all specialist ICT businesses employ less than 20 people (Houghton 2003, p. 1).

Industry statistics related specifically to software development businesses are not gathered by the Australian Bureau of Statistics (ABS). The ABS classifies software development within the Computer Consultancy Services industry. This industry classification comprises consultancy and professional services related to hardware and software, systems analysis design and programming, building of custom-designed systems, data processing and preparation services, and microfiching and microfilming services (ABS 2004b). At the end of June 2003, there were 14,500 business enterprises in the computer consultancy services industry; these businesses employed 63,500 staff and reported a total turnover of A$9.8 billion. The five major players in the computer consultancy industry are IBM Australia Ltd, EDS Australia Pty Ltd, CSC Australia Pty Ltd, MYOB Ltd and DMR Consulting Pty Ltd (IBISWorld 2004).

It has been estimated that as at December 2001, the Software and Services sector employed approximately 61,000 staff (Whitehorse Strategic Group Ltd 2002), and contributed $2.6 billion per annum to the Australian economy (SEA National 2002). However, these figures underestimate the size of the software sector as they do not include many of the firms employing less than 100 staff.

SEA estimate there are approximately 650 software development companies each with revenue of over $1 million per year and that 70 percent of these companies employ less than 20 people (SEA National 2004). There are also thousands of small firms with revenues of less than $1 million (McKerlie Consulting 1996).

The statistics available from the ABS do not take into account the substantial activity and contribution related to in-house software development and embedded software (Framework for the Future Mapping Working Group 2002). It has been estimated that the cost expended by organisations on in-house software development is equivalent to the revenue from systems and utilities software, applications tools, applications solutions and software consulting services combined (McKerlie Consulting 1996, p.20).

Software development in Queensland

The economy of the State of Queensland is comparable in size to the economy of Singapore, New Zealand or Malaysia (IIB 2000, p. 1). ABS figures for 2002/2003 report that 14 percent of all Australian computer consultancy establishments are located in Queensland (IBIS 2004). As far as the number of staff employed, data for Queensland is not published, however it is likely that the profile is similar to the Australian distribution with predominately small businesses and a few large businesses (IIB 2000, p. 3).

Estimates of the number of software businesses in Queensland vary. Excluding retailers and resellers, it is estimated that there are approximately 2,300 ICT firms engaged in developing and marketing their own products and services, particularly in software, and 22 percent of the 3,700 Queensland ICT businesses supply software development products and services (Qld Dept of State Development 2004, pp. 8-9). Earlier estimates for 1999/2000 stated that 70 percent of the 1,728 ICT firms supply software applications and systems; 33 percent provide software development services; and 32 percent provide systems design, development and implementation IT services (IIB 2000, pp. 8-9). The Queensland ICT industry recorded revenues of $14.6 billion in 2000/2001 (Qld Dept of State Development 2004) with sales for 1999/2000 of software (applications and systems) estimated at $587.8 million (IIB 2000, p. 12).

1.3 Research problem and research questions

Although Glass has declared the software crisis has passed (1996, p. 11), there are still reports of abandoned projects and software errors causing problems (Yardley 2002). As software becomes increasingly important to all aspects of industry, there is a need to encourage practitioners to adopt best practice so as to improve the quality of the processes in use, and therefore achieve targets relating to time, budget and quality. Furthermore, the local development industry needs to adopt world-class standards to be competitive (Howarth 2004a).

This presents a challenge in terms of determining the current practices of industry participants, and in devising improvement initiatives which are feasible for these very small organisations. In many small software development organisations, processes are not defined and chaotic modes of operation eventuate (Batista & Dias de Figueiredo 2000).

Currently, the level of adoption of best practice among local software developers is unknown. To help improve the software industry, it is necessary to determine the current status of use of practices and techniques. Furthermore, as the local industry is dominated by small firms, the effectiveness of assessment-based SPI for small organisations needs to be evaluated.

Thus the research problem addressed in this study is:

> ***To what extent are best practice techniques used by Queensland software development organisations, and how effective are assessment-based SPI programs for small software development firms?***

Essentially, the position taken is that there is wide variation in the adoption of software development practices, and that low-cost mini-assessments are effective for software process improvement for small development firms, including those firms with low capability.

Three specific research questions are investigated to explore the research problem. The first question is answered with data from an industry-wide survey:

> ***Research question 1.*** *Is there wide variation in the extent of adoption of software development best practice techniques by software developers in Queensland, and is adoption related to particular organisational characteristics?*

The remaining two research questions are investigated by analysing assessment reports from field experiments involving 22 small software development firms:

> ***Research question 2.*** *Is the capability of technical processes higher than that of management processes, and is process capability associated with particular organisational characteristics?*

> ***Research question 3.*** *Are assessment-based SPI programs an effective means to improve process capability for small software development firms, and is the extent of improvement associated with particular organisational characteristics?*

At this stage, the preliminary research model can be visualised as depicted in Figure 1-1.

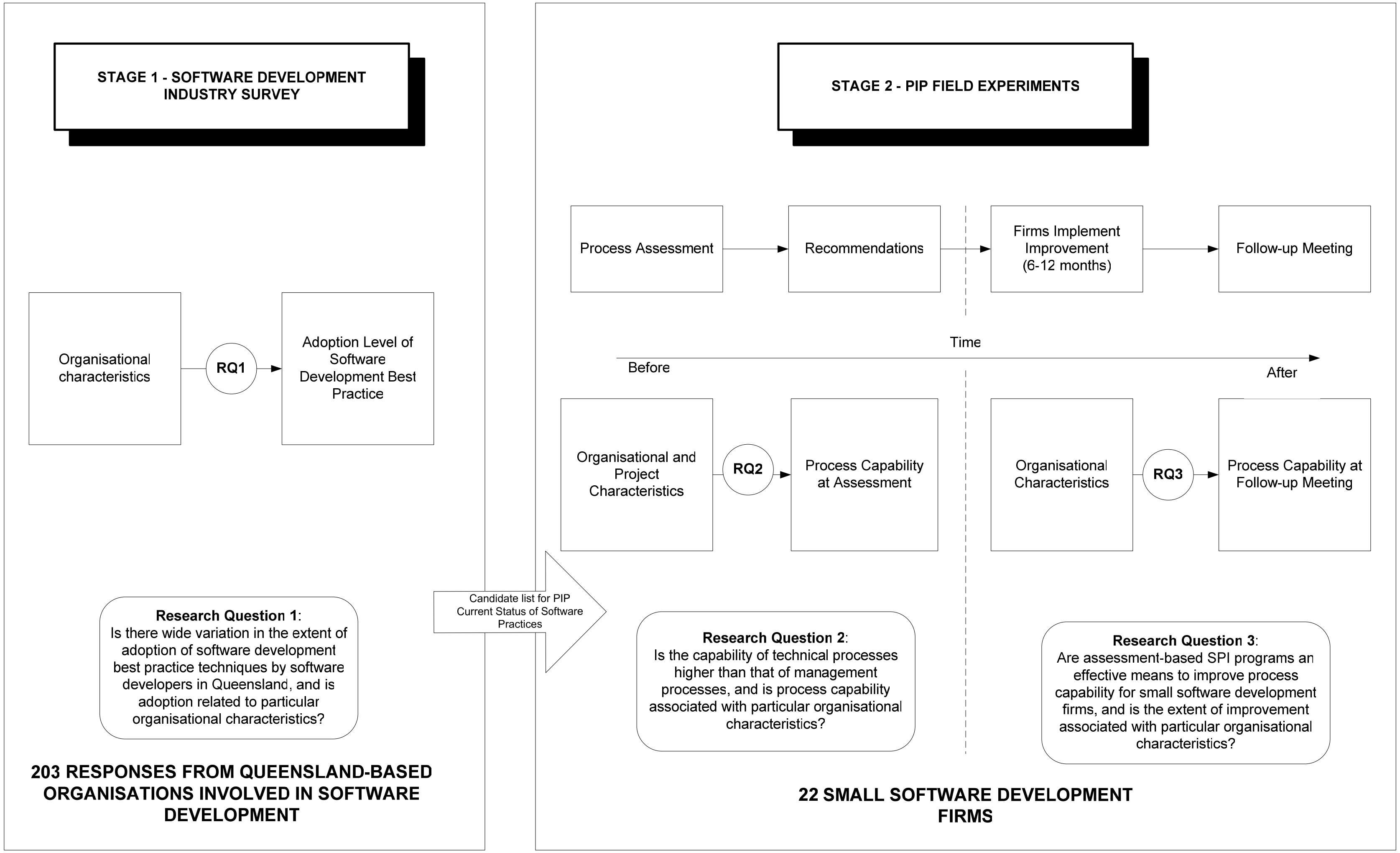


Figure 1-1 Preliminary research model

This research investigates the adoption and use of software process improvement initiatives by software development firms. In particular, it determines the use of best practice techniques across a broad range of software development organisations in Queensland, and then evaluates the impact of an assessment-based SPI initiative in 22 firms. As the software development industry is dominated by small firms, the study focuses on to small-medium Queensland software development firms.

1.4 Justification for the research

This research is justified on three grounds: the need for an industry-wide survey of current software practices; the need to use and evaluate software process improvement models; and the lack of empirical research about software process improvement within small software development firms. Each of these issues is now discussed.

1.4.1 The need for an industry-wide survey

There is much prescriptive advice written about what software engineers should be doing but rarely is any description provided about what they are actually doing (Glass 2003b). Paulk, Goldenson and White (2000, p. 1) also note the need for a survey to gain a 'good feel for the breadth of deployment of specific techniques across industry'. Ludewig (2001) considers that broad investigations on the actual state of software engineering are rarely done because 'collecting the information is a huge effort, while few people appreciate the results' (p. 1). Consequently, there are few empirical studies reported of actual software processes used (Bandinelli et al. 1995; Dutta, Van Wassenhove & Kulandaiswamy 1998). Prior to instigating process improvement programs, it is imperative to have an understanding of current software development practices throughout the industry. The best practice survey highlights areas of weakness in need of improvement. Furthermore, the survey identifies characteristics of organisations associated with adoption of best practice.

Government and industry groups also require current information about practices. The need for an industry-wide survey was highlighted in 1997, in a federal government report about the future of the ICT industry. The report recommended to 'support the collection and dissemination of improved industry statistics and undertake regular benchmarking' (Goldsworthy 1997, p. 6). The federal and Queensland state Governments provide resources to improve the software development industry. A survey such as this is essential to ensure policies relating to ICT are based on a factual representation of the industry. The findings from this research may assist government policy-makers in formulating industry improvement programs, research grants, funding for industry support organisations such as SEA and the Information Industries Bureau (IIB); and purchasing policies to help the local software development industry compete for government spending and also internationally.

A further justification for the survey is that it provides a list of candidates for the process improvement program. Therefore this research provides a valuable contribution by investigating the level of adoption of software best practice techniques across the software development industry in Queensland.

1.4.2 SPI adoption and evaluation

Process improvement is recommended to software development organisations as a means to improve effectiveness in terms of cost, schedule, quality, performance and to enhance competitiveness (Ibanez & Reo 1998; Yamamura & Wigle 1997).

In 1995, the Federal Government commissioned a study to determine the needs of the Australian software development industry. Firstly, the report identified five technology and market trends influencing the software industry: process improvement; standards; programming languages and tools; outsourcing; and packaged software (McKerlie Consulting 1996, p. 5). Software process improvement promises enhanced manageability and productivity, reduced faults, and more effective reuse, by predicting project costs and durations, meeting quality standards, capturing organisational knowledge and improving work processes (McKerlie Consulting 1996, p. 14-5). Following on from the trend analysis, the McKerlie report identified the need for accurate information on cost effective methods associated with process improvement, and the requirement for the 'independent evaluation of process improvement tools and methods' (McKerlie Consulting 1996, p. 37).

Faced with an enormous choice of methods, tools and techniques, software development managers need evidence that their investment in new practices will produce benefits (Fenton, Pfleeger & Glass 1994; Wood et al. 1999). Unfortunately, many approaches are adopted 'based on anecdotes, gut feelings, expert opinion and flawed research, not on careful, rigorous software engineering experimentation' (Fenton, Pfleeger & Glass 1994, p. 87). Therefore, researchers are urged to undertake evaluative research involving realistic projects with sufficient rigour to ensure that any benefits identified are clearly derived from the concept in question (Fenton, Pfleeger & Glass 1994, p. 87). Although past studies have indicated factors which inhibit adoption of SPI, empirical research on software process innovation is largely lacking. Consequently, there is insufficient knowledge about which innovations are effective, and which factors influence their adoption. It is vital to understand the processes currently used, and to evaluate the effectiveness of process improvement programs, or investments in SPI are wasted (Mustonen-Ollila & Lyytinen 2003, p. 275). This research provides evidence of the outcomes of software process innovation in 22 software development firms.

Assessment-based SPI programs are based on formal frameworks and promote the use of systematic processes and management practices for software engineering (Dutta, Lee & Van Wassenhove 1999). These approaches identify best practices for the management of software engineering and when applied, enable organisations to understand, control and improve development processes. The purpose of an SPI assessment is to compare the current processes used in an organisation with a list of recommended or 'best' practices, thereby identifying areas to be improved.

The Capability Maturity Model (CMM) is the foundation for many assessment-based SPI programs. These programs were designed for large organisations undertaking extensive projects. However, while this model has been evolving over many years, the software industry has changed dramatically. In the late 1960s, the main source of large scale software was development contracts issued by the U.S. Department of Defense (DoD), and it is claimed that 'since then, virtually all software engineering literature has concentrated explicitly or implicitly on the model of DoD contract software development' (Fayad, Laitinen & Ward 2000, p. 115). Today, the software development contract effort for large government departments is dwarfed by commercial off-the-shelf (COTS) software, much of which is produced by small companies (Graham & Mowery 2003). Fayad, Laitinen and Ward (2000) raise the point that issues such as company size, development mode (contract versus COTS), development size (program size, shipped volume) and development speed have not been adequately addressed in the software engineering literature. Therefore, this research addresses that need by including organisation and project characteristics in terms of how they relate to process capability, and also the extent of improvement.

This research answers the call to reduce the scepticism and uncertainty which exists in relation to the accuracy and usefulness of software process assessments and the improvements based on them (Goldenson et al. 1997). Furthermore, although there are many published accounts of assessments, there is little reported about reappraisals or follow-up assessments except for large high maturity organisations (Goldenson & Herbsleb 1995).

1.4.3 Lack of research about SPI for small firms

The lack of theory-based empirical research pertaining to SPI adoption has been noted, but nowhere is the research shortage more acute than in relation to small software development firms.

In the Australian and international software development industry, there is a large proportion of small software development firms. Such firms are involved in producing commercial-off-the-shelf packages and also custom writing software applications for clients. Although the customised software market is still substantial, its growth is being outstripped by that for packaged software, with many companies choosing to buy packaged software which they can customise internally, rather than buying a fully customised system (McKerlie Consulting 1996).

There have been many calls to recognise the importance of small business, to increase the attention given to the small business sector, and to develop government policies appropriate to the needs of small business (Johns, Dunlop & Sheehan 1989). It is also recognised that to date, business research in general, and software process improvement research in particular, is biased towards large corporations (Attewell & Rule 1991, p. 301), and that empirical research into the rate and success of implementation of process improvement initiatives in small and medium enterprises is largely considered to be inadequate (Xydias-Lobo & Jones 2003).

Recent research has raised doubts about whether traditional SPI models are appropriate for small software development organisations. This study responds to demands for more research to evaluate the effectiveness of assessment-based SPI programs within small development firms (Brodman & Johnson 1997a; Kautz 1998b).

1.5 Methodology

The underlying research paradigm of this study is positivist and the research approach is both descriptive and evaluative. The extent of adoption of best practice by software development organisations is evaluated, as is the SPI experience of a selected group of 22 firms. This study conforms to the nomothetic research style as it uses a survey and field experiments to test hypotheses within the scientific tradition. Quantitative analysis, focusing on statistical analysis of numerical data, and also qualitative analysis of textual and numerical data, are employed. As detailed in Chapter 3, this approach is consistent with the traditional research approach adopted by software engineering researchers.

A survey is considered to be a feasible means of providing data for any study investigating the state of practice (Wilson, Petocz & Roiter 1995). In this case, the survey provides a broad industry-wide snapshot of the adoption of best practice techniques in use throughout software development groups in Queensland. The mail survey is complemented by the field experiments, which provide an in-depth analysis of the outcomes of a software process improvement program involving 22 small software development firms. This type of multi-method approach is strongly advocated by Morrison and George (1995) and other software engineering and information systems researchers (for example Gable 1994; Gallivan 1997; Groves et al. 2000). It is claimed that a superior piece of research can be achieved by combining the main strength of survey research, generalisability, with the main strength of experimentation (Gutek 1991). The multi-method approach used in this study provides richer data and the opportunity to compare the best practice survey results with the results from the process improvement program.

The statistical methods used to analyse the data are based on descriptive and correlational analysis, including both parametric and non-parametric methods appropriate for the types of variables in the hypotheses, and the distribution of data.

A flowchart outlining the research process adopted for this study is shown in Figure 1-2 and full details of the methods employed are provided in Chapters 4 and 6.

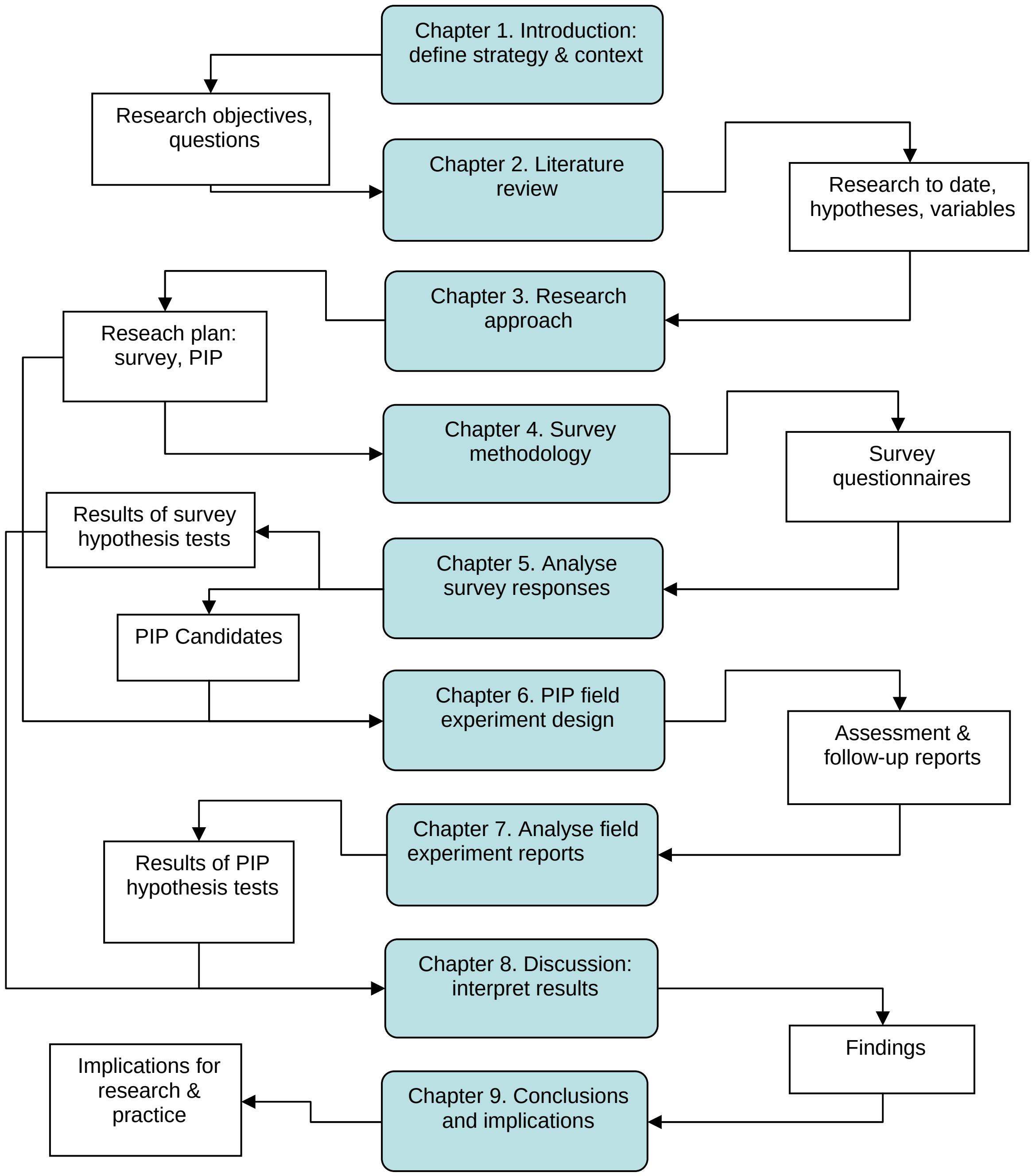


Figure 1-2 Research process flowchart showing chapters and interim work products

1.6 Outline of the thesis

This first chapter of the thesis provides a brief background of the software development industry in Queensland. The concepts of best practice and process improvement are introduced, the research problem is stated, the justification of the research presented and the methodology introduced. Delimitations of the scope of the research and key assumptions are discussed.

In Chapter 2, literature relating to the underlying theories of process improvement and diffusion of innovation is reviewed. Current research about software process improvement is summarised, highlighting the gap in research relating to the adoption of SPI by small organisations. Finally, the literature is used to formulate hypotheses to explore the research questions posed in Chapter 1.

Chapter 3 details the methodology used, firstly describing the research paradigm, approach and method. The two data collection methods, the survey and field experiments, are introduced and justified.

Chapter 4 describes the procedures followed for the best practice survey. The design and pre-test of the questionnaire is explained, followed by the selection of organisations to be included in the survey. Variables sourced from the survey are defined and the data analysis techniques identified. Limitations of the survey methodology are discussed, and ethical issues related to the survey are considered.

In Chapter 5, the data analysis from the survey responses is reported. A demographic profile of respondents is presented, followed by the analysis of the survey data, with particular emphasis on exploring the distribution of the variables representing the extent of adoption of best practice, and their association with organisational characteristics. Finally, the results of the hypothesis tests relating to the first research question are summarised.

Chapter 6 describes the procedures associated with the process improvement field experiments. The field experiments are detailed in terms of the selection of subjects, the design and protocol of the field experiments, and the compilation of results. The variables are defined and data analysis method described. Finally, limitations to the field experiment methodology and ethical considerations related to the process improvement program are discussed.

Chapter 7 focuses on the results of the field experiments, and reports the findings from the 22 assessments and also the follow-up meetings. Firstly, the 22 firms which participated in the process improvement program are summarised in terms of their organisation and project characteristics. Statistical tests are presented exploring the relationship between the process capability levels and organisation and project characteristics to test the hypotheses relating to research question 2. The extent of process improvement from the time of the initial assessment to the follow-up meetings is analysed to test the hypotheses formulated to explore research question 3. Finally, the data from the survey is compared with that gathered through the field experiments.

Chapter 8 discusses the findings in terms of each of the three research questions. The outcomes of the hypothesis tests related to the best practice survey (presented in Chapter 5) and the process improvement field experiments (detailed in Chapter 7) are discussed. This chapter provides context and meaning to the study by comparing the results with similar studies, and by qualitative analysis of the survey responses and PIP reports. Critical success factors identified in previous software process improvement studies are considered to highlight factors of particular relevance to small development firms. The results from the best practice survey are then compared with those from the process improvement program.

Chapter 9 summarises the study and presents the conclusions about the research problem and the contribution of the research to the body of knowledge. Implications of the research for theory, practice and policy are considered. The limitations of the research method and analysis are discussed and areas for future research suggested.

1.7 Definitions

Definitions adopted by researchers are often not uniform, and this is certainly the case with some of the terms used in this study. In this section, key and controversial terms are defined to establish the positions adopted in this research. For the convenience of the reader, a list of abbreviations is included as Appendix A of this thesis, and a glossary of statistical tests and terms in Appendix B.

Practice refers to a software engineering or management activity that contributes to the creation of the output (work products) of a process or enhances the capability of a process (ISO/IEC TR 15504-9 1998, p. 3).

Best practices are the most successful solutions or problem-solving methods that have been developed by a specific organisation or industry and are widely recognised as excellent, and recommended by most practitioners and experts in the field (ESI 1997; Laudon & Laudon 2004, p. 316).

Software process is a set of activities, methods, practices and transformations that people use to develop and maintain software and their associated products (Graydon, Nevalainen & Drouin 1998, p. 76).

Process capability is the ability of a process to achieve a required goal (ISO/IEC TR 15504-9 1998).

Process capability level refers to a point on an ordinal scale (of process capability) that represents the increasing capability of the performed process; each level builds on the capability of the level below (ISO/IEC TR 15504-9 1998, p. 4).

Process improvement is the operation of putting in place measures to strengthen processes which have been identified as sources of defects or risks to quality, cost or schedule performance. Process improvement is based on the premise that product quality is highly dependent upon the processes used in its creation (ISO/IEC JTC1/SC7 N944R 1992).

Process assessment is the disciplined examination of the processes used by an organisation against a set of criteria to determine the ability of those processes to perform within quality, cost and schedule goals (ISO/IEC JTC1/SC7 N944R 1992; Smith et al. 1994). A process assessment may be conducted by or on behalf of an organisation to understand the state of its own processes, or by one organisation to determine the suitability of another organisation's processes (ISO/IEC TR 15504-9 1998, p. 10).

Process improvement program is defined as all the strategies, policies, goals, responsibilities and activities concerned with the achievement of specified improvement goals (ISO/IEC TR 15504-9 1998, p. 10). In the context of this thesis, the process improvement program (PIP) refers to the activities related to the one-day RAPID-based assessments and follow-up meetings conducted during 1999 and 2000 by the Software Quality Institute (SQI).

Definition of small business

Another term relevant to this study is SME: small and medium sized enterprise. As noted by Xydias-Lobo and Jones (2003, p. 2) literature related to SMEs sometimes suffers from methodological limitations such as unclear or inconsistent definitions of what constitutes an SME. The Small Business Coalition (SBC) has called for a clear and uniform definition of small business to enable greater clarification of eligibility criteria for government programs and for the collection of more reliable statistics (SME E-Commerce Roundtable - Thinktank 2001).

Currently, there are a variety of definitions used by researchers and government agencies based on the number of employees, turnover, sector, or ownership structure (Goode 2001; Holmes & Gibson 2001; Rigby & Trantom 2004). The OECD and United States of America (US) Government define an SME as an organisation employing less than 500 staff (NZ Ministry of Economic Development 2001), and the European Commission (EC) also defines business size based on the number of employees: micro business 1-19 staff; small business 20-99; and medium business 100-249 employees (O'Regan & Ghobadian 2004). In Hong Kong, a manufacturing SME can have up to 100 employees whereas a non-manufacturing enterprise is regarded as an SME only if it has less than 50 employees, and in Mexico, industrial enterprises are classed as medium with up to 500 employees, but only need 100 employees to be classed as medium in the commercial and services sectors (Rigby & Trantom 2004). Although definitions vary, the overriding consideration in classifying organisation size appears to be based on the number of employees (Goode 2001; O'Regan & Ghobadian 2004, p. 65).

During 1999, the ABS conducted a review of the way businesses should be defined by size. In summary, the review recommended that for statistical purposes, small businesses (excluding agricultural businesses) should be defined on the basis of full-time equivalent (FTE) employment (ABS 1995). Furthermore, small organisations are defined by the ABS (2002) as businesses employing less than 20 people. The ABS recognises three categories within that definition: non-employing businesses (sole-proprietorships and partnerships without employees); businesses with one to four employees; and businesses with between five and 19 employees. The first two categories are sometimes referred to as *micro-businesses*. Organisations with 20-199 employees are classed as *medium sized* businesses. This study conforms to the ABS definition, but care is taken in comparisons involving studies which have used different definitions of organisation size.

1.8 Delimitations of scope and key assumptions

The study was limited in terms of the geographic location of the organisations studied, and also the instruments used to collect the data.

The question of access is the principal constraint in selecting a research setting (McGrath, Martin & Kulka 1982, p. 50). The researcher's association with the SQI provided opportunities for collaboration with the Queensland branch of SEA. As SEA sponsored both the survey and the PIP, the survey was conducted in the State of Queensland and the process improvement program was conducted mostly in Brisbane with one regional firm included. Although the survey only includes organisations based in Queensland, some of these firms may be the Australian headquarters for national or international development firms. Many of the surveyed organisations would have clients from all over Australia, as well as international clients. Therefore, although these organisations are based in Queensland, they are not developing software solely for the Queensland market.

For both the survey and PIP stages of the study, pre-existing instruments were used as the basis for the data collection. The European Software Institute (ESI) developed a software best practices questionnaire and this instrument was used as the basis of the survey reported here (ESI 1995). A key assumption is that the set of items making up the ESI questionnaire constitute best practice. As discussed in Chapter 2, the list of best practices included in the ESI questionnaire was derived from accepted software process improvement models.

The process improvement program reported and evaluated in this thesis was conducted using the Rapid Assessment for Process Improvement for software Development (RAPID) model and method (Rout et al. 2000). The RAPID method is based on the emerging international standard for software process assessment ISO/IEC 15504 (SPICE). The ISO/IEC 15504 standard has been validated through an international series of trials. The RAPID method is SPICE-compliant and restricted to assess eight processes: requirements elicitation, software development, project management, configuration management, quality assurance, problem resolution, risk management, and process establishment. A key assumption is that the one-day RAPID assessment provides an accurate measure of the capability of the processes under review.

1.9 Conclusion

This chapter has laid the foundations for the thesis. It introduced the research problem and research questions. The research was justified, the methodology was briefly described and justified, the thesis was outlined, definitions were presented, and the delimitations given. On these foundations, the thesis will proceed with a detailed review of literature relating to the underlying theories of process improvement in general and software process improvement in particular.

2 CHAPTER TWO—LITERATURE REVIEW

2.1 Introduction

Chapter 1 introduced the research problem: *To what extent are best practice techniques used by Queensland software development organisations, and how effective are assessment-based SPI programs for small software development firms?* The purpose of this chapter is threefold: firstly, to review literature relating to the underlying theory of process improvement; and secondly, to summarise current research about software process improvement. From this literature review, it becomes apparent that there is little research relating to the adoption of SPI in small organisations. The third purpose is to identify and define the gap in the literature and to formulate hypotheses to provide answers to the research questions posed in Chapter 1.

An overview of the chapter is now provided. Firstly, §2.2 examines the underlying theories of process improvement and provides a brief history of Total Quality Management (TQM). The management and economic theories that support TQM concepts are discussed, and empirical studies determining the value of TQM are reported. The TQM-related concepts of process improvement and organisational maturity are then introduced. Theory of diffusion of innovation is also used to provide a theoretical framework to understand the adoption of process innovations such as best practice and software process improvement. In §2.3, the focus narrows from general discussion about process improvement in organisations to a more specific view of process improvement related to software development. Two popular SPI models, CMM and SPICE are introduced. Empirical studies of best practice adoption by software developers are reported in §2.4. Following this, §2.5 summarises research about SPI costs and benefits, and SPI adoption enablers and inhibitors. In §2.6, the need for empirical research about the adoption of SPI by small software firms is demonstrated with a summary of current studies relating to small software development organisations. The research model is presented in §2.7, and the theoretical support for this model is discussed. The research questions previously introduced are extended and the hypotheses presented. Finally, §2.8 summarises the research model and supporting research questions.

Figure 2-1 presents an overview of this chapter, graphically showing the reference and immediate disciplines. To enhance readability, Appendix A includes a list of all abbreviations used in the study.

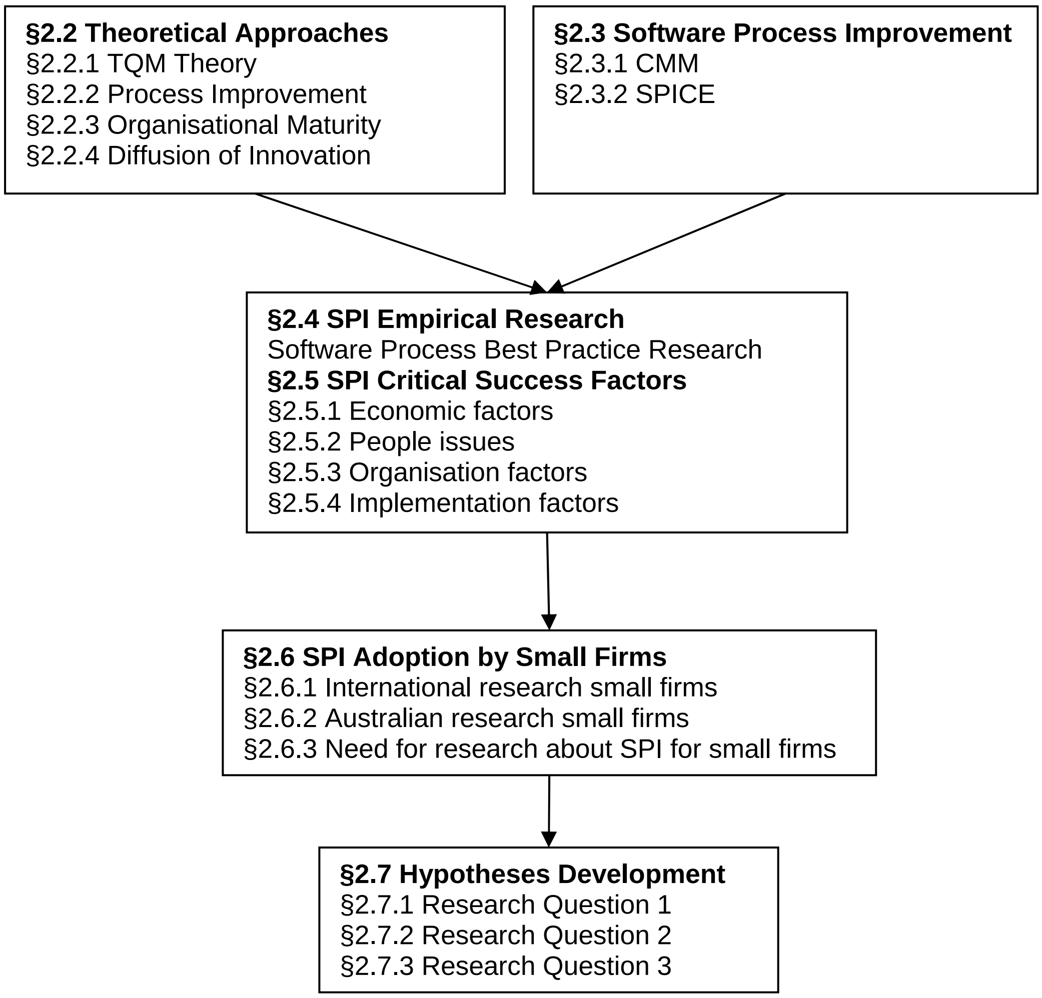


Figure 2-1 Literature review map

Need for theoretical approach

The link from theory to hypothesis is essential, otherwise, 'empirical results cannot contribute to a wider body of knowledge' (Kitchenham et al. 2002, p. 724). Further reasons are provided by Robey and Zmud (1992) who expressed concern about an observed trend towards descriptive studies unsupported by theory. They provide three reasons why theory should be used to inform research: theory offers general predictions that can be tested in specific situations, giving a logical system of hypothetical connections among general concepts; theory improves communication between researchers by offering a standardised language to describe complex phenomena, allowing researchers to learn from each other; and finally, to become and remain credible, research must be based in theory and must apply appropriately rigorous research methods to address applied questions. They go on to posit that 'Theory enables explanation and prediction. The most likely sources of general theoretical arguments about organizational phenomena are the various theories of organizational structure and process' (p. 14). Robey and Zmud (1992) also argue that mature research perspectives drawn from organisational theories, such as diffusion of innovation, provide insights into the phenomenon and increase the richness of a researcher's theoretical imagination and empirical strategies.

Researchers are urged to use robust theories and models from social sciences, economics, management and organisations to study software engineering in the context of organisational and institutional practices (Agresti 1993a; Jeffery 1993; Kling et al. 1992). The use of reference disciplines outside the Software Engineering field for theory and concepts is deemed to provide an important basis for software research fields: for example, manufacturing concepts apply to process maturity models (Glass, Vessey & Ramesh 2002).

Accordingly, in the next three sections, management and economic theories which underpin SPI are discussed.

2.2 Theoretical approaches to process improvement

Researchers have used a variety of theories and concepts from many disciplines to explain the concepts related to SPI. For example, diffusion of innovation theory (Moore & Benbasat 1991; Rogers 1995) was employed by Kaltio and Kinnula (2000), Larsen and Kautz (1997) and Mustonen-Ollila and Lyytinen (2003). Other theories applied include evaluation theory (Ares et al. 2000); motivation theory (Baddoo & Hall 2002b); marketing theory (Kaltio & Kinnula 2000); Mintzberg's organisational theory (Larsen & Kautz 1997); knowledge management theory (Meehan & Richardson 2002); organisational behaviour literature (Abrahamsson 2001); social psychology and technology acceptance and innovation adoption (Agarwal & Prasad 2000). Gray (1998) used grief theory from psychology to model resistance to SPI. Baddoo and

Hall (2002a) borrowed the repertory grid technique from the field of psychoanalysis to explore the roles of practitioners in companies which had implemented SPI.

In considering the reference disciplines for this study, TQM was chosen because Humphrey (1990), a pioneer of SPI, credits Deming's TQM concepts (Deming 1986) as motivating his Capability Maturity Model. The next section traces the history of TQM, provides a definition of TQM, and summarises empirical research reporting TQM's contribution to organisational performance.

2.2.1 TQM theory

In 1931, the American physicist, Walter Shewhart drew on the disciplines of engineering, statistics and economics to develop a concept of statistical process control (SPC) (Shewhart 1931). Shewhart believed that total quality improvement could be achieved by organisations if they followed the Shewhart cycle: plan, do, study, act (PDSA). The structure of the PDSA cycle was based on Shewhart's belief that the key to the evolution of a successful enterprise required constant evaluation of management practices and the willingness of managers to adopt ideas (ASQ; Skymark).

Shewhart's concepts were extended and refined by W. Edwards Deming, Joseph Juran and Kaoru Ishikawa to form the total quality management (TQM) approach. During the last 50 years, there have been many variations in TQM implementations, but all recognise that the key strategy of most organisations is to improve product and service quality, and continuously improve business processes (Dale et al. 2001). Researchers in the fields of engineering, economics, production, management and operations research have studied the development, application and assessment of TQM techniques (Wruck & Jensen 1994). Although TQM has been criticised as lacking underlying theories (Chiles & Choi 2000), and TQM researchers have also been criticised for not applying TQM standards to research design, measurement and analysis (Hackman & Wageman 1995), a plethora of articles and books have been published on the topic.

A review of the TQM literature reveals that controversy exists about its definition and value to organisations (Wruck & Jensen 1994). For the purpose of this study, TQM can be defined as 'an approach to management that has evolved from a narrow focus on statistical process control to encompass a variety of technical and behavioural methods for improving organisational performance' (Dean & Bowen 1994, p. 396). Although much of the TQM literature is written by consultants, there is a body of empirical research evaluating the effect of TQM on organisational performance. This is examined in the next section.

Effect of TQM on organisational performance

It has been reported that process-oriented improvement techniques have been very successful in some organisations, but have failed in many others (Repenning 2000). For example, although fewer than 100 of the 1,000 companies surveyed by Easton and Jarrell (1998) had ongoing TQM programs, those that had implemented TQM 'significantly outperformed their competitors' (Repenning 2000). From the analysis of survey responses from 62 small/medium Australian firms, Anderson and Sohal (1998) found a number of significant positive relationships between TQM practices and organisational performance. The International Quality Study was conducted jointly by the American Quality Foundation and Ernst and Young. The survey, which included over 580 organisations on three continents, concluded that only three of the 945 management practices examined have a significant impact on performance. These three practices are process improvement methods, strategic plan deployment and supplier certification (Adam 1994; Anderson & Sohal 1998).

In the US, the General Accounting Office (1991) found, from a survey of 20 companies, firms which had adopted TQM experienced an overall improvement in corporate performance, greater customer satisfaction, increased market share and improved profitability. An analysis of 2,297 companies by Barron and Paulson Gjerde (1996) uncovered evidence that firms adopting TQM practices experience greater growth in sales, employment and capital stock. After surveying manufacturers in New Zealand, Maani, Putterill and Sluti (1994) concluded that quality practice had a significant positive impact on many performance measures. Powell (1995) found from his study of 54 firms, that TQM can produce economic value to the firm, but not for all adopters. He also reports Forker's (1996) study of 65 manufacturing firms: Forker concluded that quality helps a firm to gain a competitive advantage by satisfying customer needs. In a comparative analysis of TQM and ISO 9000, Zhang (2000) interviewed managers of ten manufacturing companies in the Netherlands. This research showed that TQM has much better effects on overall performance than ISO 9000, reducing quality costs by more than seven percent.

Therefore, it is clear that TQM can provide benefits to organisations and this may account for the opinion that interest in, and efforts toward TQM are worldwide and growing rapidly (French & Bell 1995).

2.2.2 Process improvement

Regardless of the particular flavour of TQM implemented, process control and improvement is always included as it is a core TQM principle (Hackman & Wageman 1995). The main idea behind process control is that organisations are sets of interlinked processes and that improvement of these processes is the foundation of performance improvement (Dean & Bowen 1994). The importance of the focus on work processes is stressed because TQM practitioners assert that product and service quality is determined most of all by the quality of the processes by which they are designed and produced. Theory to support these concepts can be found in research from industrial engineering (Taylor 1911), and management (Hammer & Champy 1993) fields.

Benchmarking

One way to improve processes is by benchmarking whereby an organisation compares its practices with those of an organisation considered to use best practice (Dean & Bowen 1994). Following the comparison, improvements are identified and implemented. Management of process quality includes benchmarking and assuring the quality of suppliers' products and services Organisations benchmark to gain knowledge about industry best practice and then learn how they can improve the efficiency of their own work processes and ultimately better satisfy their customers with improved products and services (Chiles & Choi 2000). Rather than rely on staff members or managers for ideas for improvement, benchmarking activities may provide opportunities to imitate entirely new routines being used at other organisations, or to tap a rich source of new ideas for improving existing routines. However, caution is advised as a number of researchers have found that organisations that depend mainly on processes identified by external benchmarking, experience problems in gaining internal acceptance of these processes (Hackman & Wageman 1995, p. 317).

Process definition

Many organisations recognise the value of specifying, articulating and defining the processes in use. As well as reducing the wasted effort and subsequent cost through coordination losses and misdirection, specifying current procedures provides a means of organisational learning (Hackman & Wageman 1995). The advantages of articulating a routine are also stated by Chiles and Choi (2000) as providing a 'framework for predicting consequences of alterations of the routine and hence opportunity to plan successful intervention' and also the 'apparatus to convert tacit knowledge (which may not be understood by the firm itself) to explicit knowledge (which it can readily grasp)' (p. 199). However, there can be disadvantages if work practices are over specified. A loss in staff motivation can result if the processes are identified, documented, diffused, and standardised to the point where workers feel they have little discretion about performing their tasks (Hackman & Wageman 1995). Also, although routines, rules and procedures provide some measure of stability and order in uncertain environments, the failure to continuously improve such institutionalised processes can lead to 'an atrophy of knowledge within organisations' (Chiles & Choi 2000, p. 198).

The effect of TQM on transaction costs

Economic theories provide a useful framework to analyse some of the risks inherent in software development and the opportunity for capability assessments to provide external governance to reduce such risks. Applying transaction cost theory, Milgrom (1988) declared that over a period of time, workers accumulate firm-specific knowledge, and if they leave, the firm incurs additional costs. TQM can reduce these costs by providing a means to efficiently create and use valuable specific knowledge throughout the organisation (Wruck & Jensen 1994). Therefore, TQM can be viewed as 'an institutional mechanism for solving "the economic problem of the firm" by efficiently co-ordinating local, specialised, partly tacit knowledge dispersed throughout the firm' (Chiles & Choi 2000, p. 200).

2.2.3 Organisational maturity

Following on from the work of TQM pioneers Deming (1986) and Juran (1988), Crosby (1979) produced the quality management maturity grid. The grid has six measurement categories: management understanding and attitude; quality organisation status; problem handling; cost of quality as percent of sales; quality improvement actions; and summation of company quality posture. For each of these categories, there are five stages of maturity: stage I uncertainty; stage II awakening; stage III enlightenment; stage IV wisdom; and stage V certainty (Crosby 1979, pp. 38-9). Crosby encouraged managers to use the grid to assess the current situation and to identify actions needed for improvement.

Earlier, Likert (1967) had defined four distinct stages of organisational maturity which he referred to as *systems of organisation*: system 1 exploitive authoritative; system 2 benevolent authoritative; system 3 consultative; system 4 participative group. His research, conducted with the use of the now popular Likert-type questionnaires, found that system 4 organisations are the most effective and successful whereas system 1 organisations face many problems. Tully, Kuvaja and Messnarz (1999) traced the history of staged maturity models from Plato's four stage ascent of the mind, through Marx's four stages of society development and Rostow's five stages of economic growth and concluded that 'stage models, whether of philosophers, economists, quality gurus, or software engineers, can be seen as occupying a respectable place in that utopian tradition' (Tully, Kuvaja & Messnarz 1999, p. 56).

Since Crosby's work, maturity models have gained popularity and have been proposed for a range of activities including quality management, software development, supplier relationships, research and development effectiveness, product development, innovation, product design, product development collaboration and product reliability (Fraser, Moultrie & Gregory 2002). As well as the CMM, described in §2.3.1, another well known and often applied IT model is Nolan's stages of growth (Nolan 1973) which considers the adoption steps of IT within organisations.

Having established that process innovation is part of TQM, and that the notion of organisational maturity is rooted in TQM practice, the next section introduces diffusion of innovation theory to provide a framework to understand the complex phenomenon of the dissemination of process innovations throughout an industry.

2.2.4 Diffusion of innovation

Diffusion of innovation theory is a rich research area which has been used by many researchers to analyse the adoption of best practice techniques and SPI by software developers (Agarwal & Prasad 2000; Bayer & Melone 1989; Fichman 1992, 1994, 1995; Mustonen-Ollila & Lyytinen 2003; Zmud 1982).

An *innovation* is defined as any idea, object, or practice that is perceived as new by members of the social system (Mahajan & Peterson 1985). The *diffusion of an innovation* has been defined as the process by which that *innovation* is communicated through certain *channels* over *time* among the members of a *social system* (Mahajan & Peterson 1985; Rogers 1995). Relating three of these four key elements in the diffusion process to this study:

- the *innovation* is best practice techniques, in other words, software development processes which are widely recognised and recommended by practitioners and experts
- the *social system* is software developers in Queensland
- the *time* period relates to the study time period, January 1999-December 2000.

It is recognised that innovations can be either adopted from external sources or transferred from internal sources by learning from and formalizing best practice (Mustonen-Ollila & Lyytinen 2003), but temporal and financial constraints precluded the examination of *channels* of communication in the study reported in this thesis.

Rogers' (1995) principles of the diffusion of innovation are as follows: the role of communication influences perception of innovation; appropriate knowledge is necessary; suitable communication channels are important; change agencies and agents play a significant role; potential adopters expect economic compensation (rewards); and networks supported by mentors are needed. The model proposed by Rogers (1995) depicted the rate of adoption as a normal distribution curve rising as the proportion of adopters increased from the innovators and early adopters to the early majority. At this point, the rate of adoption peaked and then the late majority and laggards also took up the innovation. This model has been modified by Moore (1999) who claimed that for the adoption of high technology innovations, the curve is not continuous since a chasm exists between the early adopters (the technology enthusiasts and visionaries) and the early majority (the pragmatists). To cross the chasm and thereby ensure successful adoption, Moore believes that the core technology must be accompanied by policies, training, reference materials, procedures, systems integration, job aids, tooling and installation support (Graettinger 2003). Although Moore's theory has been promoted by researchers at the Software Engineering Institute (SEI) as a useful model to understand the adoption of the CMM and CMMI (CMM Integration), to date it has not been supported by empirical evidence.

In the 'modern software practices' study, Zmud (1982) distinguishes two types of innovations: those primarily serving the interests of the technical core and the other serving the administrative core. He also differentiates between product and process innovations: '... product innovations refer to the introduction of new products or services that shift or expand an organisation's domain, while process innovations refer to the introduction of new methods, procedures or responsibilities within existing domains' (Zmud 1982, p. 1424). Using Zmud's classifications, SPI can be classified as a *process* innovation serving both technical and administrative cores.

2.2.5 Conclusion to referent discipline theory

While TQM theory provides an understanding of process improvement and organisational maturity, diffusion of innovation theory provides a framework to understand the adoption and diffusion of process innovations. Diffusion of innovation research suggests that the adoption of best practice and SPI can be studied in terms of the innovation itself, the social system, the relevant time period and channels of communication. This basic framework has been extended to recognise the influence of change-agents, such as government purchasing bodies. The next section will move from the broad management, operations research, engineering, manufacturing literature of TQM and diffusion of innovation to focus on SPI research literature.

2.3 Software process improvement

Many software projects have been plagued by quality problems evidenced by unmet requirements and schedule and budget overruns. For over three decades, efforts have been made to improve the quality of software (Azuma 1995). Initial efforts were directed at coding, testing and debugging. Then software engineering emerged as a discipline, introducing and institutionalising techniques such as structured analysis and design. These techniques were applied to the early stages of the software development lifecycle, and provided tools to enable requirements to be specified and designs to be documented. However, these activities were time consuming, and documentation was often neglected (Dunn & Ullman 1994). Various CASE tools were then developed to facilitate diagram construction, and to check consistency through the design, library and configuration control, outputs and reporting (Smith 1995). Other methods claimed to ease the 'software crisis' include object oriented design and programming, formal methods, and software fault tolerance (Fenton 1993a).

The inability to effectively manage software development and maintenance has been exacerbated by the growing size of projects, and a '... cultural tradition rooted in the prowess of the individual programmer' (Curtis & Paulk 1993, p. 381). Various project management techniques have been deployed as it became recognised that estimates, schedules, budgets and reviews are just as important as programs, and that coordination of resources becomes more critical as project size and complexity increase. Consequently there are an extensive variety of techniques in use, as evidenced by the results of recent surveys (Cusumano et al. 2003; Davis et al. 1992; Glass 2003b). These studies reported the use of a wide range of development techniques and methods, covering project management, specification and design, coding, testing, and peer reviews.

Consideration of this diversity of techniques highlights problems related to documentation standards and training of analysts and programmers. As if this broad range of practices reported above is not problematic enough, in their UK survey to determine the extent of use of structured methods, Hardy, Thompson and Edwards found that '... only 44% of respondents reported using a recognised structured method or using formal specifications' (1995, p. 471). The most common type of methods used by organisations in their sample were '... peculiar to particular departments' (1995, p. 471), and it was also claimed that 'only about half of all organisations actually follow a methodology' at all (Riemenschneider, Hardgrave & Davis 2002, p. 1135). Recently, an Australian survey found that many companies (60%) claimed to be implementing SPI but almost half of them could not name the SPI model they used (Sweeney Research 2003a).

To overcome inconsistencies in methods and practice, Curtis and Paulk (1993) posit that many organisations are now looking to software process improvement programs to first focus on defining the processes used, and only then selecting tools and methods to support these processes. They claim a well-defined and documented process can provide an underlying foundation for long-term productivity and quality growth by integrating people, tasks, tools and methods.

In response to the recognition of the value of process assessment and improvement in relation to software development, a variety of assessment-based programs have been formulated and implemented. The first to be widely known was the Capability Maturity Model (CMM), which is discussed in §2.3.1. While the CMM became popular in the USA, the software industries in other countries were developing, using and sharing locally-grown models such as TickIT (UK), Bootstrap (Europe), Trillium (Canada), ISO 9000-3 and AMI to develop quality management systems or improve software processes.

Within the international software engineering discipline, it became apparent that there was a need to harmonise these various approaches and to develop an international standard. The standard became known by its acronym *SPICE*, and is now known as ISO/IEC 15504. SPICE is discussed in §2.3.2.

Software process improvement models such as CMM and ISO/IEC 15504 are based on formal frameworks and promote the use of systematic processes and management practices for software engineering (Dutta, Lee & Van Wassenhove 1999). These approaches identify best practices for the management of software engineering and when applied, enable organisations to understand, control and improve development processes. The purpose of an SPI assessment is to compare the current processes used in an organisation with a list of recommended or 'best' practices. In the UK, Baddoo & Hall (2002b) found that many companies now have either a formal or informal SPI program based on one of the popular SPI models such as CMM or ISO/IEC 15504.

2.3.1 Capability maturity model

In 1985, Ron Radice and others at IBM adapted Crosby's quality management grid for the software industry (Radice et al. 1985). This work was directed by Watts Humphrey who took the concepts from IBM to the Software Engineering Institute (SEI). Watts Humphrey recognised that this approach could help the US Government overcome problems with poor performance of software subcontract companies. Consequently, the SEI developed the Capability Maturity Model for software (SW-CMM) based on the organisational and quality management maturity models developed by Likert (1967) and Crosby (1979) respectively (Kan 2003).

The aim of the SW-CMM is to identify the key management and development practices necessary to produce quality software (Humphrey 1988). The term capability maturity has caused some confusion as it mixes two related concepts: organisational maturity and process capability. Processes become more capable when they are standardised across the organisation and their performance is monitored against historical data. This enables variation in performance to be detected, and ultimately, the process should be continuously improving through identifying the root causes of variability and innovative ways to fulfil its objectives (Ahern, Clouse & Turner 2001). Improvements in process capability can lead to organisational maturity, whereby goals of cost, schedule, functionality and product quality are able to be achieved (Paulk et al. 1995).

The SW-CMM has influenced software engineering practice, not only in the USA, but in many countries. It has been said that it is 'impossible to exaggerate the influence of one man [Watts Humphrey] and his model on the field of software process analysis and improvement' (Tully, Kuvaja & Messnarz 1999, p. 52). Although it has now been superceded by the CMMI, a brief discussion of the SW-CMM is included here, prior to introducing the CMMI.

CMM for software

The SW-CMM rates organisations on a five-level scale of organisational maturity as initial, repeatable, defined, managed or optimised, and also provides organisations with guidance for measuring software process maturity and establishing process improvement plans. The CMM describes the key elements of an evolutionary improvement path for software organisations from an ad-hoc, immature process to a mature disciplined one. The CMM covers practices for planning, engineering, and managing software development and maintenance (Paulk et al. 1995). As many software development organisations in the USA have embarked on process improvement programs as a means of attaining higher quality, lower development and maintenance costs, shorter time to market and increased predictability of product and process, a defacto quality indicator began to emerge for capability assessment (Kitson 1996; Paulk 1995; Rout 1992). As well, there have been criticisms of the CMM, that it is incomplete or flawed, and that it encourages too much bureaucracy (Bollinger & McGowan 1991; Brodman & Johnson 1994; Drehmer & Dekleva 1993; Fischer, Achterberg & Vinig 1993; Fitzgerald & O'Kane 1999; Herbsleb & Goldenson 1996).

A report from the US Data and Analysis Centre for Software presents empirical evidence to support claims that software process improvement can reduce development and maintenance costs, improve customer satisfaction, reduce cycle time, increase profitability, and improve professional staff (McGibbon 1999). An analysis of 40 US Air Force projects found that cost and schedule performance improved with increasing process maturity (Lawlis, Flowe & Thordahl 1995). In addition, many intangible benefits are reported: improved developer morale, increased respect for software from users, and less required overtime (Brodman & Johnson 1995). The SEI claims that higher maturity levels are associated with lower risk, higher productivity and higher quality of the software process (Humphrey, Kitson & Kasse 1989). There have been many published reports and case studies claiming dramatic improvements from the application of CMM-based software process improvement programs in the United States (Basili & Green 1994; Henry et al. 1994; Herbsleb et al. 1994; Humphrey, Kitson & Gale 1991; Johnson 1994; Jones 1996; Kitson & Masters 1993; Paulk, Humphrey & Pandelios 1992; Sims 1994; Thomas & McGarry 1994; Wohlwend & Rosenbaum 1993). In particular, documented success stories present a credible case for CMM-based assessments resulting in improvements in relation to cycle time, defect density, and productivity (Billings et al. 1994; Diaz & Sligo 1997; Dion 1992, 1993; El Emam & Birk 2000b; Humphrey, Snyder & Willis 1991; Paulk et al. 1995; Pitterman 2000; Wohlwend & Rosenbaum 1993, 1994; Yamamura & Wigle 1997).

The adoption of the SW-CMM by the US Department of Defense to assess its process improvement activities and the use of capability assessments in awarding software contracts accelerated the take-up of SW-CMM by US software development firms (Yamamura & Wigle 1997). Acceptance followed in other countries as a result of subsidiary software development firms of USA companies and large US-based multinational firms using the SW-CMM to assess local software companies in the UK, India, Australia and other countries.

CMM integration

A suite of models developed by the SEI including the SW-CMM, the Systems Engineering Capability Maturity Model, and the Integrated Product Development Capability Maturity Model have recently been merged and extended into an integrated CMM called CMMI. The CMMI team's mission included the objective of ensuring that all of the products developed are consistent and compatible with ISO/IEC 15504 (CMMI Product Team 2002). The CMMI provides two views of capability: a staged view and a continuous view. The staged view gives five levels of evolution towards *organisational maturity* (initial, managed, defined, quantitatively managed, and optimizing). The continuous view provides six levels of *process capability* (incomplete, performed, managed, defined, quantitatively managed, and optimizing (CMMI Product Team 2002, p. 18-9).

The SEI claims the CMMI model improves upon the best practices of previous models in many important ways (Goldenson & Gibson 2003). CMMI best practices enable organisations to do the following:

- link management and engineering activities more explicitly to business objectives
- expand the scope of and visibility into the product lifecycle and engineering activities to ensure that the product or service meets customer expectations

- incorporate lessons learned from additional areas of best practice (e.g., measurement, risk management, and supplier management)
- implement more robust high-maturity practices
- address additional organisational functions critical to its products and services
- more fully comply with relevant international standards such as ISO 9000 and ISO/IEC 15504 (Chrissis, Konrad & Shrum 2003; CMMI Product Team 2002).

2.3.2 ISO/IEC 15504 SPICE

The British Ministry of Defence was responsible for the ImproveIT project which recommended the formulation of an international standard for software process assessment (ISO/IEC JTC1/SC7 N872 1991). The Software Process Improvement and Capability dEtermination (SPICE) project was established as a project to develop draft standards, trial the developing standard and to promote awareness of the developing standard (Rout 1996). Many current SPI approaches from researchers and practitioners in over 20 countries have been incorporated to develop a consistent and validated framework for assessment and improvement (Thomson & Mayhew 1997). The output from the SPICE project team is the emerging international standard on software process assessment ISO/IEC 15504 which has undergone extensive validation through a series of trials (El Emam & Birk 2000a; Jung et al. 2001).

SPICE trials

The first phase of the SPICE trials took place in 1995. The first phase was based on the Software Process Assessment report (SPICE Project 1995) and included 35 trials (SPICE Project Team 1998). A later survey of 14 of the SPICE phase 1 trial organisations indicated that many organisations struggle with achieving successful SPI based on process assessments (SPICE Project 1998, p. 164). Recommendations and conclusions from the trials contributed to the review and rework and helped the standard evolve to the provisional draft technical report (PDTR) version.

Recently, a concentrated research effort associated with the second phase of the SPICE trials has seen the publication of many analyses of the data from the trials. The following reports are based on use of the PDTR version of the standard which was published in 1996 (ISO/IEC PDTR 15504 1996): summaries of 63 assessments (El

Emam 2002; Jung et al. 2001); validating the software requirements analysis process against productivity in 56 projects world-wide (El Emam & Birk 2000a); analysis of difficulties and benefits (Hunter 1999); validating project management (Hwang & Jung 2003); the regional factor-European data different to elsewhere (Hunter & Jung 2000); process capability, ISO 9000 and organisation size (Jung & Hunter 2001); and Australian organisations characteristics and process capability (Rout, Tuffley & Hodgen 1998).

The next stage of the development of ISO/IEC 15504 saw the release in 1998 of the technical report (TR) version of the standard (ISO/IEC TR 15504 1998). This version was used as the basis of the process improvement program evaluated in this study. Further development has since occurred and the four parts of the new five-part international standard have now been published (Rout 2003). In the remainder of this document, in order to enhance readability, the term *ISO 15504* is used to refer to ISO/IEC TR 15504.

SPICE reference model

There are two dimensions of the ISO 15504 reference model, the process dimension and the process capability dimension. The process dimension includes measurable objectives for each process and relates to the software lifecycle model (ISO 12207). The process dimension is made up of 24 main processes and 16 sub-processes as summarised in Appendix D, Table D.4. The process capability dimension is measured by capability levels. ‘A capability level is characterized by a set of attribute(s) that work together to provide a major enhancement in the capability to perform a process. Each level provides a major enhancement of capability in the performance of a process. The levels constitute a rational way of progressing through improvement of the capability of any process’ (ISO/IEC TR 15504-2 1998, p. 4).

The ISO 15504 technical report sketches out a roadmap for the implementation of best practice in software engineering by defining 40 processes, divided into five categories: customer-supplier; engineering; support; management; and organisation. The process capability of each defined process evaluates to what extent the process achieves its defined purpose and objectives (SPIRE 1998). For each process, capability is measured in six levels from incomplete through performed, managed, established, predictable to optimising. These capability levels represent milestones along the road to software process improvement. There is growing interest in the emerging standard: it has been estimated (by El Emam & Garro 2000) that approximately 1,260 SPICE-based software process assessments were conducted during the 22 months from September 1996 to June 1998.

Summary

This summary of the history of software process improvement explains the current situation which is characterised by a wide variety of practices and methodologies, and the emergence of CMM and SPICE as standards for best practice. In the next section, research is reviewed relating to organisations' adoption of best practice and specifically, empirical research, which has evaluated the use of assessment-based SPI models such as CMM and SPICE.

2.4 Empirical research relating to best practice and SPI

Over the last 10 years, interest in SPI has attracted the interest of researchers and practitioners as evidenced by the growing number of publications about SPI in well-respected journals such as *IEEE Software, IEEE Transactions on Software Engineering, Communications of the ACM, Journal of Systems and Software, IBM Systems Journal, Annals of Software Engineering, Software Engineering Journal, Empirical Software Engineering Journal,* and *Software Quality Journal*. Additionally, a journal is published dedicated to the topic, the *Software Process Improvement and Practice Journal*. Also, research about software process improvement is reported at international and national conferences such as IEEE International Conference on Software Engineering, Australian Software Engineering Conference, Asia-Pacific Software Engineering Conference, Software Quality Management Conference, and the International Conference on Software Methods and Tools. Conferences are also held specifically focused on SPI, for example the International Conference on SPI, European Conference on SPI, European SEPG, and the International SPICE Conference.

Much of this research has emanated from software engineering research centres such as the International Software Engineering Research Network, Software Quality Institute (Australia), Software Engineering Institute (Ma USA), Fraunhofer Institute for Experimental Software Engineering (Germany), European Software Institute (Spain), Center for Software Engineering (Ca USA), Centre for Empirical Software Process Research (UK), and the Centre for Software Engineering (Ireland). The next section reviews research related to software process best practice.

Software process best practice research

Although many authors refer to software developers using dominant, prevalent, or common practices, there has been little research to date to document actual current use. A survey conducted in 1992 found marked differences in the practices used by 102 European firms compared to 326 Japanese companies (Azuma & Mole 1994), but Blackburn, Scudder and Van Wassenhove (1996) concluded from a study of software management practices in US, Japan and Western Europe that globally, firms appear to be remarkably similar in the practices used. Other researchers have focused on a particular location, for example, a survey of adoption of software processes of 280 companies in Singapore (Tan & Yap 1995); large US projects (Curtis, Krasner & Iscoe 1988); quality practices in the UK (Davis et al. 1992); and requirements engineering (El Emam & Madhavji 1995). There have been studies in Australia as well; these are summarised in §2.6.2.

It is a complex undertaking to compile a universally accepted list of best practice techniques. There are a number of well known software process models such as SPICE, CMM, ISO 9000 and Bootstrap. These models were analysed by Wang and others (Wang et al. 1997; 1998) and the superset of 444 practices covered by these four models were distilled to 83 base process activities. The 444 practices were drawn from the 210 base practices of SPICE, SW-CMM's 150 key practices, the 177 management issues in ISO 9000, and Bootstrap's 201 quality system attributes.

At the same time as the survey reported in this study was being conducted, other researchers were investigating the state of practice in software development (Cusumano et al. 2003; Glass 2003b; Groves et al. 2000; Jones 2003; McCaffery et al. 2004; Neill & Laplante 2003; Ng et al. 2004; Weber & do Nascimento 2002). The findings from these best practice studies are compared with the findings from this study in the discussion in Chapter 8.

The most widely reported survey of best practice in Europe was that conducted by the European Software Institute (ESI) (Dutta, Lee & Van Wassenhove 1998; Dutta, Van Wassenhove & Kulandaiswamy 1998; ESI 1995, 1996a). The ESI survey is discussed in the next section. The ESI survey is replicated in this study to explore the extent of adoption of best practice by software development organisations in Queensland.

European software best practice surveys

In 1995, the European Commission (EC) launched the European Systems and Software Initiative (ESSI) program with the aim of motivating organisations to test and deploy software best practices. The ESSI program was administered in coordination with the ESI as part of the EC's Information Technologies program (Dutta, Lee & Van Wassenhove 1999). Organisations were encouraged to apply for EC funding to enable them to adopt a specific process improvement project in a real-life commercial environment over a period not exceeding 18 months (Kautz 1998a). These SPI projects were known as Process Improvement Experiments (PIE) and their results were collated and stored in the PIE repository (ESI). The program included a longitudinal study of European software practices to assess and monitor the level to which European software developers were adopting best practices.

The ESI developed the Software Best Practices Questionnaire (SBPQ) to collect data for the ESSI program. Previous research in software process improvement and popular models such as the CMM, Bootstrap and ISO 15504 influenced the development of the questionnaire. On three occasions (between 1995 and 1997), the questionnaire was distributed by the ESI as part of the call for proposals for ESSI funding. Respondents were explicitly informed that the questionnaire was independent of the funding proposal review process (Dutta, Lee & Van Wassenhove 1999).

A best practice is defined as 'a management practice that is widely recognised as excellent and is recommended by most practitioners and experts in the field' (ESI 1997). The SBPQ represents the 'subjective consensual views of multiple experts' (Dutta & Van Wassenhove 1997a), and comprises a subset of core software development practices extracted from recognised SPI models.

The ESI survey instrument groups practices under five headings:

- organisational issues

  - project management, change control, training programmes for managers
- standards and processes
  - formal assessment of benefits and risks, management reviews, control of subcontractors, independent audits, coding standards, formal handovers, test planning
- metrics
  - records of actual and estimated resources, error sources, test efficiency, computer performance, project tracking
- control of the development process
  - accountability for estimates and schedules, requirements management, control of code and specification changes, regression testing
- tools and technology
  - use of design notations, automated testing tools, prototyping, data dictionary, project management tools.

The content of the questionnaire has been criticised on two counts by Dutta and Van Wassenhove (1997a): firstly, it overlooks important issues related to organisational and customer-supplier management; and secondly, it does not including practices associated with high maturity organisations (for example, CMM level four and five practices).

However, despite its shortcomings, the ESI study yielded valuable findings from the analysis of the 1,279 responses received over three years. There were 463 responses to the first survey in March 1995 from 17 countries and representing 33 different sectors (ESI 1996a). The second survey was conducted in mid 1996 and received 488 responses from 17 countries representing 34 different sectors (ESI 1996a).

The last survey in 1997 generated 397 responses (ESI 1997) and showed ‘wide variation in both awareness and application of process improvement techniques’ (Dutta, Lee & Van Wassenhove 1999). Overall, the respondents had adopted 51 percent of all practices. Developers from sectors exhibiting low adoption rates (such as mechanical engineering) were encouraged to benchmark against companies in the aircraft and spacecraft sector as it showed high levels of adoption. European organisations were urged to become more aware of best practices and process-improvement techniques as the European software industry was lagging behind the US in terms of awareness and application of SPI (Dutta, Lee & Van Wassenhove 1999).

Many of the organisations who responded to the SBPQ survey were successful in gaining EC funding to improve their software processes. Consequently, there has emerged a valuable body of single-case studies detailing the effectiveness of specific SPI initiatives in small and medium-sized companies. These studies are in stark contrast to the publications emanating from the US about large high-maturity companies. The US studies tend to focus on large firms involved in US Defense contracts, whereas the PIE reports include smaller firms from various European countries developing software for a diverse range of application domains.

2.5 SPI critical success factors

Many researchers have used the concept of critical success factors (CSFs) to identify key areas where attention must be paid to ensure success. Since Rockart (1979) introduced the concept, CSF studies have proved useful in the analysis of the adoption and use of information systems and management practices. Some studies refer to the critical success factors and critical barriers as enablers and inhibitors respectively.

There has been much research published about critical success factors for SPI adoption. The research approach varies; surveys have been used (Goldenson & Herbsleb 1995); some of the research is based on single or multiple case studies (Ares et al. 2000); some researchers analysed previously published research (Niazi, Wilson & Zowghi 2003; Rainer & Hall 2001; Stelzer & Mellis 1998); and there is much published in academic and practitioner journals which is not based on empirical research methods, but advocates approaches or provides advice to firms (Ibanez & Reo 1998; Kasse & McQuaid 1998; Saiedian & Chennupati 1999). As evidence of the substantial research activity, Dyba's (2000) review of the SPI literature derived nearly 200 prescriptions for success. In order to analyse the variety of factors addressed in SPI research to date, the framework and summary provided by Hall, Rainer & Baddoo (2002) is used here and extended to enable groupings of the empirical SPI research into four categories: studies relating to **economic** factors; research relating to **people**

issues; research about **organisational** factors; and finally, publications relating to SPI **implementation** factors. Studies relating to more than one issue may appear in multiple categories. A summary of this literature is provided in Table 2.1.

2.5.1 SPI adoption—economic factors

Despite a warning from Hersh (1993, p. 12) that it is not easy to measure the value of process improvement in terms of lower risk, productivity per staff month, improved quality, or customer satisfaction, many publications have appeared claiming to have determined the return on investment for process improvement. Recently, high costs and inadequate resourcing have been found to be the greatest impediment to SPI success (Bucci, Campanai & Cignoni 2001; Hall, Rainer & Baddoo 2002). A recent survey of Australian software organisations found that high cost was the main barrier preventing SPI adoption (Sweeney Research 2003a).

Return on investment. In §2.3.1, reference was made to many published accounts of CMM implementation returning substantial financial benefits (Billings et al. 1994; Dion 1992, 1993; Humphrey, Snyder & Willis 1991; Paulk et al. 1995; Wohlwend & Rosenbaum 1993). In the US, many organisations have reported return on investment ratios up to almost 9:1 (Goldenson & Herbsleb 1995; U.S. General Accounting Office 1991, p. 1), but these extreme claims were challenged by Fenton (1993a) who asserted that they are not backed up by hard empirical data. More recent reports have provided a convincing argument of financial benefits of software process improvement (McGibbon 1999; Solon & Statz 2002). The SEI investigated the results of CMM-based SPI efforts and deduced that the average return on investment for SPI was 5.7:1 (Krasner 1997). Brodman and Johnson (1995) found that commercial organisations focus on benefits such as improvements in productivity, quality and schedule, rather than financial return.

In contrast to the large organisations reported above, Batista & Dias de Figueiredo (2000) studied a very small team which implemented SPI and found that over 12 months, monthly costs decreased by 33 percent while monthly benefits increased by 17 percent of their monthly value. One often publicised benefit is reduction in delivered defects, for example, Yamamura and Wigle (1997) report from Boeing that

CMM helped the defect identification rate to improve from 89 percent to nearly 100 percent.

Payback period. A recent industry-based study (Debou & Kuntzmann-Combelles 2000) rated the late impact of SPI programs on projects as a very important issue. Debou and Kuntzmann-Combelles suggest that action plans from an assessment should cover a three to five month timeframe as management tends to lose patience and practitioners lose momentum when planning for longer time periods. From a longitudinal study of a major IT company, Slaughter, Harter and Krishnan (1998) evaluated the return on investment in SPI and concluded that 'larger returns from quality improvement occur early in the project'. However, Paulk et al. (1995 p. 98) and Krasner (1997) warn of late returns with process changes taking two years to demonstrate results, and Kulpa (1998) found the average time to obtain return on investment was three to five years.

2.5.2 SPI adoption—people issues

The literature reflects growing awareness about the important role of individuals in SPI programs. This is stressed by Komiyama, Sunazuka and Koyama (1998) who claim that the process determines the success of the software project, and that all personnel must be interested in the process. As small firms rely greatly on key individuals, human factors are particularly important for them (Horvat, Rozman & Gyorkos 2000).

Management commitment and SPI leadership. Management commitment and support is essential for any major process improvement. Top management can take a leadership role and adopt a longer-range perspective of the benefits thus ensuring sufficient allocation of resources and overcoming organisational resistance (Thong, Yap & Raman 1996). Management commitment was identified by Stelzer and Mellis (1998) as the most important success factor for both CMM and ISO 9000-based software process improvement efforts. In the case of Hughes Aircraft, Humphrey, Snyder and Willis (1991) believed management commitment played an important role in the success of the SPI initiative. This was echoed by Johnson (1994) in his evaluation of ISD Corning's experience. At Schlumberger, Wohlwend and Rosenbaum (1993) noted that replacing a committed manager with one with less commitment to SPI caused previous improvements to be lost. Managers and SPI leaders should be consistent, stable, and highly respected to be effective change agents (Abrahamsson 2001; Butler 1997; Debou & Kuntzmann-Combelles 2000; Dyba 2002; Goldenson & Herbsleb 1995; Herbsleb & Goldenson 1996; Quann 1997; Rainer & Hall 2003; Wiegers 1998).

Staff involvement. The importance of staff involvement, empowerment and process ownership is a basic concept of TQM. It is not surprising then that many studies have confirmed that participation is viewed as an essential requirement for successful SPI adoption (Bach 1995; Dyba 2002; El Emam, Fusaro & Smith 1999; Goldenson & Herbsleb 1995; Herbsleb & Goldenson 1996; Johnson 1994; Paulk et al. 1995; Stelzer & Mellis 1998). From their survey analysis of 85 companies in the UK, Hall, Rainer and Baddoo (2002) identified that the quality of internal SPI staff is a fundamentally important aspect of SPI success. Such people should be experienced members of SPI teams. In large companies, it is possible to set up a dedicated group to evaluate and decide on SPI issues, however, in smaller organisations, all employees expect to be involved (Kelly & Culleton 1999). SPI is reported to have contributed to an improvement in employee satisfaction from 74 percent to 96 percent (Yamamura & Wigle 1997).

Mentors. Mentoring is viewed as a means to provide leadership, motivation and training. As early as 1993, a formal mentoring program was established as part of the Onboard Shuttle project. This mentoring process shortened the apprentice period for new staff and increased productivity (Paulk et al. 1995). A single case study about establishing mentor relationships in an SPI project with a large US Army organisation showed many benefits from implementing the mentoring methodology and recognised that the achievement of CMM level 3 rating was accelerated by the mentors' involvement (Reeb & Henderson 1997). The important role of mentor is also confirmed by Kautz (1998b) and Nolan (1999).

Training and expertise. Mentors can provide on-the-job training, but formal training may also be required. An important factor raised (by Debou & Kuntzmann-Combelles 2000) involved the lack of software management skills. After reviewing 39 published reports concerning 14 high maturity sites, Rainer and Hall (2001) concluded that process expertise was one of the main issues associated with successful SPI programs. Dyba (2002) identified exploitation of existing knowledge and exploration of new knowledge as critical success factors. An effective training strategy is proposed (by Thorwart 1998): first educate internal experts and have them provide support to their colleagues.

Motivation. From a study involving 49 focus groups with 200 software practitioners at 13 UK companies, Baddoo and Hall (2002b) found the most important common motivators to SPI adoption were process ownership, evidence of SPI success, and resources for the SPI programme. Also, developers are more interested in SPI-specific motivators such as bottom-up initiatives and top-down commitment, whereas managers cite organisational-specific motivators such as meeting targets and cost benefit. Based on the same study (Baddoo & Hall 2002a), it was found that software developers often have a limited view of SPI and often do not see themselves assuming ownership of processes. Other factors which provide motivation are competitive pressure (Johnson 1994), and external regulation (Paulish & Carleton 1994).

2.5.3 SPI adoption—organisational factors

In an attempt to identify the organisational and cultural characteristics of large successful Australian IT firms, Lowry, Morgan and FitzGerald (1996) noted from their study of six organisations that all placed a high reliance on software quality evaluation and practices, with all using a standardised and documented software development methodology on projects.

In discussing the results of an assessment at a large Australian organisation, Rout (1998) highlighted the important fact that the reference model defined in ISO 15504 includes the organisational and supporting processes, not only the primary processes associated with the software lifecycle. It has also been noted that SPI needs effective change management otherwise the culture and politics within the organisation can create a barrier to SPI adoption (Herbsleb & Goldenson 1996; Kitson & Masters 1993).

Based on the European Quality Model, an empirical evaluation was carried out by Dutta and Van Wassenhove in collaboration with the ESI (1997b). This study is of particular interest because it 'integrates the strengths of software maturity and TQM models' (1997b, p. 1) by collecting data related to business performance as well as software development practices. There were 85 responses to the survey (conducted in 1995), mainly from Europe but including a handful from USA, Mexico and Russia. The results of the survey revealed that excellent organisations formally evaluate the business value of software process improvement investments, and focus on 'retaining and satisfying their customers by delivering high quality, reliable products on time' (Dutta & Van Wassenhove 1997b, p. 18).

Communication. One way to overcome resistance to SPI is to establish and maintain effective communication and feedback channels (Butler 1997). A strong communication effort is needed before and during the formal SPI implementation, and it is essential that new processes are disseminated to all team members so that best practice can be shared (Stelzer & Mellis 1998; Wohlwend & Rosenbaum 1993). More than just informing staff about SPI is required. To overcome resistance to change, the new initiative must appeal to staff affected. Therefore, Kaltio and Kinnula (2000) stress that it is critical that the SPI message is formulated to match the audience.

Resources. SPI tends to be more successful when staff time and resources dedicated to SPI are good or excellent (Herbsleb & Goldenson 1996, p. 328). This view is also held by Baddoo and Hall (2003), Brodman and Johnson (1994), Kaltio and Kinnula (2000), and Paulish and Carleton (1994). From an analysis of 85 cases in the UK, Hall, Rainer and Baddoo (2002) reported that although companies may have objectives for SPI, they often do not adequately resource their SPI programs.

Business strategy. The importance of aligning the SPI method with the organisation's goals and strategy has been stressed by many researchers (Debou & Kuntzmann-Combelles 2000; Dutta & Van Wassenhove 1997b; Dyba 2002; Ibanez & Reo 1998; Kulpa 1998; Rifkin 2001; Rout 1998). It is particularly difficult to achieve this alignment if the organisation is restructuring or changing its goals. After reviewing 39 published reports concerning 14 high maturity sites, Rainer and Hall (2001) concluded that organisational stability was one of the main issues associated with successful SPI programs.

2.5.4 SPI adoption—implementation factors

There are a variety of implementation factors which can cause the demise of a well-planned SPI initiative. Action plans are needed after the assessment, and SPI should be treated as a project (Wiegers 1998). It is also important to ensure new processes are institutionalised or developers may return to using the previous processes (Bamford & Deibler 1998; Butler 1997; Wiegers 1998; Yamamura & Wigle 1997).

Setting realistic objectives. Previously, the importance of linking SPI programs to business goals was discussed. As well, it is necessary to have realistic objectives which can be achieved in the foreseeable future, and guard against ‘long-term and fuzzy goals’ (Stelzer & Mellis 1998, p. 239). However, it is difficult to set realistic goals. Herbsleb and Goldenson (1996) found (from their survey of 138 CMM-assessed organisations) that firms recognised the importance of setting realistic expectations and making goals clear but underestimated the time and expense of SPI. This problem resulted in a fair amount of discouragement. El Emam, Fusaro and Smith (1999) surveyed 14 organisations one year after they had undertaken SPICE assessments. They found that one of the most critical success factors was that the SPI goals were well understood. A related issue is the need to tailor SPI to suit the organisation (Fitzgerald & O’Kane 1999; Saiedian & Chennupati 1999; Varkoi 2004). In 90 percent of the CMM cases analysed by Stelzer and Mellis (1998), the need for tailoring SPI initiatives was emphasised.

SPI infrastructure. According to Kaltio and Kinnula (2000) from their experience at Nokia, the critical elements for successful deployment of processes are a well-organised and institutionalised support infrastructure, the right product in terms of human enactment, and promotion of SPI through training and ‘selling’ the idea to staff. To improve the technology infrastructure at Nokia, an Intranet accessible Lotus Notes system was successfully implemented for the process library (Kaltio & Kinnula 2000).

Infrastructure encompasses human as well as IT resources. Organisations are urged to set up SPI steering groups, Software Engineering Process Groups (SEPG) or control boards made up of representatives from different projects and functions to research and promote SPI and to provide coordination on critical issues (Hall, Rainer & Baddoo 2002; Herbsleb & Goldenson 1996; Humphrey, Snyder & Willis 1991).

Evaluation. Organisations should evaluate various SPI models to find one that suits its business needs (Saiedian & Chennupati 1999). Furthermore, the SPI project should be evaluated (Paulk et al. 1995; Wilson, Hall & Baddoo 2001). Hall, Rainer and Baddoo (2002) found that companies were generally ineffective at evaluating the impact of SPI.

Readiness. A number of SPI research articles refer to the concept of readiness, but there is no standard definition for the term. To explore aspects of software process assessment and improvement, Thomson (1995) drew on Rubin's readiness footprint model which had originally been developed to assess adoption of CASE tools (Rubin 1990). Rubin's view of technology transfer dynamics proposed that an 'organisation's ability to assimilate advanced software engineering technology was a function of its innate 'readiness' footprint that encompassed eight key dimensions: motivation, investment, skills, education, culture, organisation, technology and applicability' (Rubin 1991). He hypothesised that successful technology transfer is a function of the gap between an organisation's readiness and the characteristics of technology it is attempting to assimilate (Rubin 1991). The Rubin concept of readiness is broad and includes most of the critical success factors identified by other researchers (and already discussed here).

A narrower interpretation of readiness has been taken by some researchers (Smith et al. 1994; Vakaslahti 1997; Wilson, Hall & Baddoo 2001). After analysing reports which indicate that the vast majority of organisations in the US and UK are at the initial level of capability maturity, Smith et al. (1994, p. 207) asserted that it is clear that only a handful of companies were ready for SPI 'because their software health is so bad (that is if they have any development process at all)'. They warned that in order to be ready for SPI, a visible and defined software process must already be in place. The importance of starting with defined processes was also reported (by Wilson, Hall & Baddoo 2001) in a study of evaluation and prediction of SPI success involving seven UK companies. Vakaslahti (1997) recommends the use of the personal CMM (Curtis, Hefley & Miller 1995) first to estimate organisational maturity and readiness prior to undertaking re-engineering and SPI investments. Training in the SPI model and management practices is also suggested as an appropriate process improvement entry strategy (Kasse & McQuaid 1998).

The literature reviewed above is summarised in Table 2.1. This summary does not purport to provide an exhaustive list of all research relating to SPI critical success factors. However, it does illustrate that a substantial research effort has been expended by way of surveys, case studies, and analysis of published reports. The extent and variety of research to date demonstrates that SPI adoption is a very complex issue and there is a myriad of factors which can influence the outcome of an SPI program. Researchers are unable to agree on one set of factors: there is little consensus and many issues overlap. This highlights the need to understand more about the relationship between organisation factors and adoption of best practice such as SPI. The framework for summarising the issues, adapted from Hall, Rainer and Baddoo (2002) is also used to structure the evaluation of the assessments in the discussion in Chapter 8.

Table 2.1 Summary of literature related to SPI critical success factors

Category	Factors	Research approach	Author reference
Economic	Value of SPI	Expert judgement/experience	Hersh (1993)
		Survey analysis 85 UK firms	Hall, Rainer and Baddoo (2002)
		Survey and rapid assessments Central Italy	Bucci, Campanai and Cignoni (2001)
		Focus groups and phone surveys	Sweeney Research (2003a)
	ROI	Case study Hughes Aircraft	Humphrey, Snyder and Willis (1991)
		Case study Schlumberger	Wohlwend and Rosenbaum (1993)
		Case study Tinker Air Force Base	Paulk et al. (1995)
		Case study Raytheon	Dion (1992; 1993)
		Case study IBM space shuttle onboard software	Billings et al. (1994)
		Survey of 20 companies	U.S. General Accounting Office (1991)
		Survey follow-up of 56 appraisals	Goldenson and Herbsleb (1995)
		Literature review	McGibbon (1999)
		Analysis of Gartner Application Development benchmark database	Solon and Statz (2002)
		5 case studies big US companies + summaries 23 published payoff stories	Krasner (1997)
		Survey and interviews	Brodman and Johnson (1995)
		Case study: 1 very small team	Batista and Dias de Figueiredo (2000)
		Case study Boeing	Yamamura and Wigle (1997)
	Payback period	Case studies of 2 firms	Debou and Kuntzmann-Combelles (2000)
		Longitudinal study 1 company	Slaughter, Harter and Krishnan (1998)
		Case study On board shuttle	Paulk et al. (1995)
		5 case studies big US companies + summaries 23 published payoff stories	Krasner (1997)
		Experience from CMM assessments	Kulpa (1998)
People Issues	Roles	Case study NEC	Komiyama, Sunazuka andKoyama (1998)
		Case studies 17 companies Slovenia	Horvat, Rozman and Gyorkos (2000)
	Management commitment and	Survey of 119 German firms	Stelzer and Mellis (1998)
		Case study Hughes Aircraft	Humphrey, Synder and Willis (1991)

Category	Factors	Research approach	Author reference
	SPI leadership	Case study Corning	Johnson (1994)
		Case study Schlumberger	Wohlwend and Rosenbaum (1993)
		Survey follow-up of 56 appraisals	Goldenson and Herbsleb (1995)
		Survey	Herbsleb and Goldenson (1996)
		Consultant/experience	Wiegers (1998)
		Experience – consultant/trainer	Quann (1997)
		Case study Tinker Air Force Base	Butler (1997)
		Case study Scandinavia	Abrahamsson (2001)
		15 practitioner groups at 4 companies	Rainer and Hall (2003)
		Survey 120 organisations – Norway	Dyba (2002)
		Case studies of 2 firms	Debou and Kuntzmann-Combelles (2000)
	Staff involvement	Survey 120 organisations – Norway	Dyba (2002)
		Survey of 119 German firms	Stelzer and Mellis (1998)
		Survey follow-up of 56 appraisals	Goldenson and Herbsleb (1995)
		Survey	Herbsleb and Goldenson (1996)
		Expert judgement/experience	Bach (1995)
		Case study On board Shuttle project	Paulk et al. (1995)
		Survey 14 SPICE assessed organisations	El Emam, Fusaro and Smith (1999)
		Case study Corning	Johnson (1994)
		Survey analysis of 85 companies in the UK	Hall, Rainer and Baddoo (2002)
		Experience from 1 organisation	Kelly and Culleton (1999)
		Case study – Boeing	Yamamura and Wigle (1997)
	Mentors	Case study IBM On board Shuttle project	Paulk et al. (1995)
		Case study US Air Force	Reeb and Henderson (1997)
		Action research 3 firms Denmark	Kautz (1998b)
		Case study Rolls Royce	Nolan (1999)
	Training and expertise	Case studies of 2 firms	Debou and Kuntzmann-Combelles (2000)
		Analysis of reports	Rainer and Hall (2001)
		Survey	Dyba (2002)

Category	Factors	Research approach	Author reference
		Case study German company	Thorwart (1998)
	Motivation	13 UK companies	Baddoo and Hall (2002b; 2002a)
		Case study Corning	Johnson (1994)
Organisation	Standards	Interviews 6 large Australian IT firms	Lowry, Morgan And FitzGerald (1996)
		International survey	Dutta and Van Wassenhove (1997b)
		Case study large Australian firm	Rout (1998)
		Survey	Herbsleb and Goldenson (1996)
		Analysis of data from 59 CMM assessments	Kitson and Masters (1993)
	Communication	Case study Tinker Air Force Base	Butler (1997)
		Survey of 119 German firms	Stelzer and Mellis (1998)
		Case study Schlumberger	Wohlwend and Rosenbaum (1993)
		Case study Nokia	Kaltio and Kinnula (2000)
	Resources	49 focus groups at 13 companies	Baddoo and Hall (2003)
		Survey	Herbsleb and Goldenson (1996)
		Case study Nokia	Brodman and Johnson (1994)
		Case study Nokia	Kaltio and Kinnula (2000)
		Case studies Siemens	Paulish and Carleton (1994)
		Survey analysis of 85 companies in the UK	Hall, Rainer and Baddoo (2002)
	Business strategy	Consultant/researcher	Rifkin (2001)
		Case study large Australian firm	Rout (1998)
		Case studies of 2 firms	Debou and Kuntzmann-Combelles (2000)
		Prescriptive	Ibanez and Reo (1998)
		Survey 85 responses mainly Europe	Dutta and Van Wassenhove (1997b)
		Experience from CMM assessments	Kulpa (1998)
		Survey 120 organisations Norway	Dyba (2002)
		Analysis of reports	Rainer and Hall (2001)
Implementation	Institutionalisation	Consultant/experience	Wiegers (1998)
		Case study Boeing	Yamamura and Wigle (1997)
		Case study Tinker Air Force Base	Butler (1997)

Category	Factors	Research approach	Author reference
		Consultants/assessors/trainers	Bamford and Deibler (1998)
	Realistic objectives and tailor SPI	Survey of 119 German firms	Stelzer and Mellis (1998)
		Survey 138 CMM assessed organisations	Herbsleb and Goldenson (1996)
		Survey 14 SPICE organisations	El Emam, Fusaro and Smith (1999)
		SPI assessments	Varkoi (2004)
		Framework to evaluate models – not empirical	Saiedian and Chennupati (1999)
		Longitudinal case study Motorola Cork	Fitzgerald and O'Kane (1999)
	SPI infrastructure	Case study Nokia	Kaltio and Kinnula (2000)
		Survey 138 CMM	Herbsleb and Goldenson (1996)
		Case study Hughes	Humphrey, Snyder and Willis 1991)(1991)
Evaluation	Models & project	Framework to evaluate models – not empirical	Saiedian and Chennupati (1999)
		Survey analysis of 85 UK cases	Hall, Rainer and Baddoo (2002)
		Interviews 7 UK firms	Wilson, Hall and Baddoo (2001)
		Case study IBM On board Shuttle project	Paulk et al.(1995)
	Readiness	UK case studies	Thomson (1995)
		Informal poll at conference	Rubin (1991)
		Analysis of reports/literature	Smith et al. (1994)
		Interviews 7 UK firms	Wilson, Hall and Baddoo (2001)
		27 interviews at 2 projects in 1 company – estimate readiness before large scale investment in SPI	Vakaslahti (1997)
		Entry strategies: discussion from experience	Kasse and McQuaid (1998)

As shown in Table 2.1, the size of the organisation has been identified as an important issue in SPI adoption. The next section focuses on research relating to SPI adoption by small firms.

2.6 SPI adoption by small firms

Since 1987, the CMM has been adopted by many large organisations, initially those interested in contracting to the US Department of Defense, and although glowing reports of benefits to large firms have been published, small firms have reported problems in trying to adopt the CMM. As early as 1992, Brodman and Johnson reported that the cost of CMM implementation was prohibitive for small firms (1992). They summarised problems experienced by 200 small businesses throughout the US in using CMM: 'documentation overload; unrelated management structure; inapplicable scope of reviews; high resource requirements; high training costs; lack of needed guidance; unrelated practices' (Brodman & Johnson 1997a, p. 661). This view of SPI methodologies being beyond the reach of small companies has been echoed since by many researchers (Bucci, Campanai & Cignoni 2001; Grunbacher 1997; Kautz 1998b; Larsen & Kautz 1997; Pfleeger et al. 1997; Pringle 2001; Richardson & Ryan 2001; Wilkie, McFall & McCaffery 2004). Furthermore, Fayad and Laitinen (2000) question whether the traditional SPI models, based on the US Defense scenario of large bespoke projects, are appropriate for today's smaller, newly founded software development firms which are aiming at the software package mass market.

Organisational behaviour and management literature establishes small organisations are different to larger organisations in terms of formalisation, centralisation, complexity and personnel ratios (Daft 1998). Furthermore, research has highlighted other characteristics of small firms compared to large firms:

- Small organisations have a flat structure and are managed by their owners in an organic, free-flowing, personalised management style that encourages entrepreneurship and innovation, less formalised decision-making structures and procedures, and more freedom for employees to depart from the rules (Attewell & Rule 1991; Daft 1998; Johns, Dunlop & Sheehan 1989; O'Regan & Ghobadian 2004).

- Uncertainty, evolution and innovation play a greater role in small firms (Storey 1994).
- All the critical management decisions such as finance, accounting, personnel, purchasing, processing or servicing, marketing and selling are made by one or two persons, without the aid of internal specialists, and with specific knowledge in one or two functional areas (Johns, Dunlop & Sheehan 1989).
- Small firms are averse to consultants and reluctant to seek external help (Cragg 2002, p. 277).
- The personal involvement of employees in small firms encourages motivation and commitment because the employees identify with the company's mission (Daft 1998).
- Small organisations have the advantage of being responsive and flexible (Daft 1998; Grover & Teng 1992; King 1996).
- Compared to larger firms, small firms neglect training (Voss et al. 1998).
- In small firms, much of the work is coordinated through direct supervision and mutual adjustment (Mintzberg 1983).
- Small firms have faster employment growth rates and generate more new jobs than giant ones (Attewell & Rule 1991).

Therefore, small firms should not be considered to be scaled down versions of large firms (Richardson 2002; Storey 1994), and it is clear that process improvement models such as the SW-CMM which were developed for large software contractor firms may not be appropriate for small firms. Much of the IT research to date is biased towards large corporations (Attewell & Rule 1991), and does not take into account issues relating to small organisations.

2.6.1 International research SPI small firms

The important role of small firms and the growing interest in SPI adoption is evidenced by the recent research interest on an international scale as shown in Table 2.2.

Table 2.2 Examples of international empirical research on SPI for small firms

Location	Author	Research approach	SPI Model
Austria, Ireland, Italy, Sweden	Lawthers (1999)	60 firms	SPICE
Norway	Larsen and Kautz (1997)	Survey 166 firms	Not specific. Industry-wide awareness
Ireland	Richardson (2002); Richardson and Ryan (2001)	Case studies	Software Process Matrix based on QFD
Brazil	Anacleto et al. (2004a; 2004b)	4 firms	MARES based on SPICE
Brazil	Weber and do Nascimento (2002)	Annual nationwide surveys	ISO 9000, CMM, SPICE
USA	Brodman and Johnson (1994)	Case study Nokia	CMM
Turkey	Demirors et al.(2001)	Case Study 1 firm	ISO 9001
Denmark	Kautz, Hansen and Thaysen (2001)	Case Study 1 firm	IDEAL (CMM)
Denmark	Kautz (1998a; 1998b)	Action research 3 firms	SPICE
Finland	Varkoi (2004)	22 firms	SPICE
Italy	Bucci, Campanai and Cignoni (2001)	Questionnaire based assessments	Rapid method based on SPICE
Germany	Thorwart (1998)	Case study 1 firm	Synquest assessments
Portugal	Batista and Dias de Figueiredo (2000)	Case Study 1 firm	CMM
Norway	Dyba (2000; 2002; 2003)	Survey – 120 responses from 55 companies	From literature review
Northern Ireland	Wilkie, McFall and McCaffery (2004); McCaffery et al. (2004)	Survey and 6 appraisals	APA based on CMMI
Slovenia	Horvat, Rozman and Gyorkos (2000)	17 companies	Processus SPISC

Many of the case study projects listed in Table 2.2 report on efforts to tailor existing SPI models to provide an effective solution for small software development firms:

- LOGOS tailored CMM for small business, small organisations, and small projects (Brodman & Johnson 1997b)
- Processus SPISC based on CMM and ISO 9000-3 (Horvat, Rozman & Gyorkos 2000)
- Agile process appraisal (APA) based on CMMI appraisal method (Wilkie, McFall & McCaffery 2004)

- Mini-assessments based on CMM (Natwick, Draper & Bearden 1999)
- Software Process Matrix based on quality function deployment (Richardson 2001)
- Rose Informatik used Synquest assessments (Thorwart 1998);
- MARES (software process assessment methodology) based on SPICE (Anacleto et al. 2004a, 2004b)
- SATA-Spin SPICE-based assessments (Varkoi 2004).

Although some studies have raised doubts about the effectiveness of formal SPI for small firms (El Emam 2002, pp. 6-7; El Emam & Birk 2000b; Richardson 1999), other studies have established that small firms can overcome problems experienced with formal SPI models by using a structured, tailored model (Kautz, Hansen & Thaysen 2001); clearly assigning roles and responsibilities (Kautz, Hansen & Thaysen 2001); recognising the burden placed on employees by documentation (Horvat, Rozman & Gyorkos 2000); and ensuring that the SPI program is supported by external mentors and funding (Kautz 1998b; Lawthers 1999).

It should be noted at this stage that two different research projects have used the term *rapid* and both are based on ISO 15504. The rapid method used in the Italian study (Bucci, Campanai & Cignoni 2001) focussed on three specific processes in a half-day assessment, whereas the RAPID method developed by the SQI in Australia includes eight processes with one-day assessments (Rout et al. 2000). The RAPID method was used in the research reported in this thesis to assess the process capability of small software firms, and one of the aims of this research is to evaluate the effectiveness of RAPID as an assessment-based SPI for small firms.

2.6.2 Australian research related to best practice and SPI for small firms

Prior to the commencement of the study reported here, and since its instigation, a number of reports have been commissioned by SEA National, the Queensland Government Information Industries Board (IIB) and also the Commonwealth Department of Industry, Science and Tourism (DIST) and later Department of Communications, Information Technology and the Arts (DCITA). Table 2.3 summarises the industry reports, some of which are specific to the software industry, whereas others have a broader focus of the Information and Communication Technology (ICT) Industry.

Although these reports provide useful statistics about the number of firms and employees, and revenue and expenditure in the software industry, they do not provide accurate details about the sort of practices used by software development organisations, or efforts to improve software processes. Furthermore, these reports tend to focus on large firms, although many acknowledge the existence of a large number of small software firms.

McKerlie (1996, p. 29) identified the adoption of process improvement tools as a requirement, recognising that as software development businesses grow, their initial dependence on innovation is replaced by an increasing need to use processes based on best practice. The McKerlie report also identified the need for accurate information on cost effective methods associated with process improvement, and identified that SPI adoption for SMEs was inhibited due to the complexity of process improvement tools, the prohibitive costs, lack of awareness and knowledge about process improvement tools, and the need for independent evaluation of process improvement tools and methods.

This opinion was confirmed by the *Stocktake* report in 2000: ‘In Australia, there is no definitive current information on the levels of adoption of software quality assurance and process improvement methods among developers. Nevertheless, anecdotal evidence suggests awareness and adoption among small-to-medium enterprises is very low’ (James 2000).

Further detail about best practice of Australian developers was provided in the Benchmark report (SEA National 2001). Although the Benchmark report claims to provide a view of the state of software development practices in Australia, it acknowledges that it only includes data from projects of firms with ‘sufficient process maturity to include a software metrics program’, and in fact, it reflects the ‘best’ 25 percent of the industry (SEA National 2001, pp. 4-5). The Benchmark study of 737 projects found that apart from the use of application generators and methodologies, the

219 Australian projects compared favourably with projects from Europe (227 projects) and the Americas (252 projects).

Since the study reported in this thesis commenced in 1999, awareness of SPI by developers has increased. In 2003, SEA (National) released the results of a national study into the attitudes and opinions of the software industry. The study used a multi-method approach which incorporated focus groups, interviews and telephone surveys and involved respondents from senior management to programmers, targeting SEA subscribers (Sweeney Research 2003a). Small organisations (3-19 employees) were well represented in this study (35% of respondents). Two of the seven major findings of this study relate to SPI: firstly, 64 percent of respondents considered SPI to be critically important to their business; and secondly, although 60 percent of companies claimed to be implementing SPI, almost half of these SPI adopters were unable to name the methodology in use. Furthermore, the cost of adopting and implementing SPI was ranked as the fifth major concern of software professionals (Sweeney Research 2003a). SPI models mentioned by respondents included ISO 9000, CMM and CMMI, and SPICE.

Table 2.3 Summary of Australian industry reports relating to software development

Title	Date	Contents/focus	Source of data	References
Needs Analysis for Enhancement for Software Development Capability within Australian Industry	1996	DIST: determine the needs of software industry and how these needs might be met to grow the software development industry	Interviews with 30 organisations	McKerlie Consulting (1996)
Software Development in South Australia	1996	To understand the state of the software engineering industry in South Australia (SA)	Interviews 76 SA companies engaged in software development	ASEI (1996)
Queensland IT&T Industry Survey	1996/7	Qld IT growth, revenue, target industries, employment	Survey	IIB (1997)
The Global Information Economy: The Way Ahead	1997	National Information Industry Strategy	Consultation with information and user industries, state and federal governments	Goldsworthy (1997)
Stocktake of Australia's Information Industries	1998	Strengths, weaknesses, opportunities and threats. Prepared for the Information Industries and Online Taskforce	Analysis and review of previous reports and other available material	James and Hont (1998); James (2000; 2001)
The Contribution of the Packaged Software Industry to the Australian Economy	1998	Investigation of structure and economic linkages of the software industry in the region.	Survey of BSA members and published reports	BSA and PricewaterhouseCoopers (1998)
The IT Engine Room: SMEs in Australia's IT& T Industry	1999	DCITA: Issues, challenges and opportunities for IT&T industry to promote, establish, nurture and grow	Focus groups and survey	Benson, Bull and Standen (1999)
Queensland ICT Industry Profile	2000	ICT industry employment, revenue, markets, future outlook	Mailed questionnaire—878 responses	IIB (2000)
SEA Australian Software Industry Benchmark 2001	2001	A comparison of software development projects from Australia, Europe and the Americas; and a view of the state of software development practices in the Australian software industry	Analysis of ISBSG data repository of software projects	SEA National (2001)

Title	Date	Contents/focus	Source of data	References
Towards a Software Coast: discussion paper	2001	Key challenges facing Australian software industry	Interviews and survey	Boston Consulting Group (2001); Dale, Gillmour and Maltz (2002)
Multinational Enterprise Benchmark Survey	2001	Assess the health of the Australian software and systems engineering industry; trends in critical metrics.	Surveys of 6 multinational firms 1996-2001	SEA National (2001)
Software Industry Report	May 2002	ICT industry employment, revenue and research	Survey Dec 2001	Whitehorse Strategic Group Ltd (2002); SEA National (2002)
AusStats: 8126.0 Information Technology, Australia.	Sept 2002	Australian Bureau of Statistics	Survey June 2001	ABS (2002)
Software Industry Research Report	Dec 2002	ICT industry employment, revenue and research	Survey	Whitehorse Strategic Group Ltd (2002); SEA (2003)
An Overview of the Australian ICT Industry and Innovation Base	2002	The ICT industry, its relationship to the Australian and global economy and its current capabilities	Reports	Framework for the Future Mapping Working Group (2002)
Australia: Winning in the Global ICT Industry	2002	Strategic business perspective of the ICT industry in Australia in a global context	Interviews	McKinsey and Co. (2002)
What's Bugging the Australian Software Industry	2003	Investigation of attitudes and perceptions of software development professionals	Focus groups and survey	Sweeney Research (2003b; 2003a)

As well as the industry reports summarised in Table 2.3, there have been a number of Australian academic publications focussing on the adoption of best practice and SPI, as summarised in Table 2.4. Wilson, Petocz and Roiter (1995) investigated the inconsistency in adoption of software quality assurance by comparing management's perception of use of practices with the actual use as reported by developers. The following organisational and cultural characteristics of leading Australian IT firms were identified by Lowry, Morgan and FitzGerald (1996): staff stability and longevity of service; preoccupation with technical quality; and importance of nurturing relationship with clients. Jeffery and Zucker found a weak relationship between metrics adoption and the size of the IT group, and a concentration on project management rather than process improvement metrics (1997).

Due to the involvement of SQI staff Terry Rout (as Project Editor, ISO 15504) and Angela Tuffley (regional SPICE trials coordinator), many Australian organisations have participated in the SPICE trials. A summary of 33 assessments from 15 Australian organisations from phase 2 of the SPICE trials shows 14 percent of processes rated at level 0, 32 percent at level 1, 25 percent at level 2, 28 percent at level 3 and 1 percent at level 5. This result indicates that some Australian firms can be rated with world best practice, but generally need to improve metrics collection (Rout, Tuffley & Hodgen 1998).

A limited number of case studies of SPI adoption have also been undertaken. After considering SPI adoption in four organisations, Rout, Neilson and Gasston (1994) concluded that it was important to augment the quantitative approach of assessments with qualitative research methods. Rout (1998) examined the importance of the capability of organisation processes in a large company, especially in relation to strategic planning, and Scott et al. (2001) reported a trial of an SPI framework tailored for small firms.

Table 2.4 Australian academic empirical research: best practice and SPI 1994-2000

Research focus	Data collection method	Authors
Comparison of espoused and actual software quality practices	Survey—responses from 13 organisations	Wilson, Petocz and Roiter (1995)
Processes used by successful IT firms	6 firms	Lowry, Morgan and FitzGerald (1996)
The state of practice in the use of software metrics	Survey to ASMA members, responses from 42 organisations	Jeffery and Zucker (1997)
Experiences with different approaches to software process assessment	Case study 4 firms	Rout, Neilson and Gasston (1994)
Effectiveness of software processes	Case study	Gasston and Rout (1994)
Evaluating improvement strategies using software process assessment	Case study – large organisation	Rout (1998)
Results from Australian participants in SPICE phase 2 trials	Assessments from 15 Australian organisations	Rout, Tuffley and Hodgen (1998)
Framework for SPI for small firm–IMPACT project	Case study 1 small firm	Scott et al. (2001)
SPICE for small organisations	Case study 1 firm	Tuffley, Grove and McNair (2002)

2.6.3 Need for research into SPI for small firms

A common theme emerging from the international research, as well as the Australian industry studies, is the need for research about SPI adoption by small firms. It has been noted that large and resourceful organisations report their experience but little is known about software process improvement in small organisations (Kautz 1998a). It is considered necessary to investigate how SPI approaches can be modified to better suit the predominantly informal structure of smaller organisations (Larsen & Kautz 1997, p. 85).

As noted in §1.2.2, the Australian software industry is dominated by small firms and makes a substantial contribution to the Australian economy. Therefore, it is important to ascertain whether organisations are using best practice, and to then evaluate an assessment-based SPI program which is tailored for small development firms.

2.6.4 Conclusion

As shown in Figure 2-1, a hierarchical structure was used for the literature review for this research. Firstly, relevant theories were sourced from the fields of organisation and management, namely TQM, process improvement, organisational maturity, and diffusion of innovation. Prior research pertaining to SPI was then summarised, with particular emphasis on CMM and SPICE process assessment models. Empirical studies of software process best practice were reviewed, followed by a summary of research relating to critical success factors associated with SPI. The focus was then narrowed to report on research specific to small firms and their adoption of SPI. In the next section the hypotheses are formulated.

2.7 Hypothesis development

This section describes the development of the research model for this study. Following advice from Webster and Watson (2002), the reasoning for hypotheses comes from three main sources: theoretical explanations for 'why'; past empirical findings as summarised in the previous sections; and practice or experience. As the 'why' or logical reasoning is the most important component of the explanation, it is part of the justification for all hypotheses.

The research hypotheses are presented in the natural or alternative form to enhance readability and clarity, but in each case, the null can be inferred from the stated hypothesis as positing no relationship, and it is the null that will be tested in Chapters 5 and 7, not the alternative form (Sproull 1988).

The three research questions were introduced in Chapter 1 (§1.3) and are restated here for the convenience of the reader.

Research question 1: Is there wide variation in the extent of adoption of software development best practice techniques by software developers in Queensland, and is adoption related to particular organisational characteristics?

Research question 2: Is the capability of technical processes higher than that of management processes, and is process capability associated with particular organisational characteristics?

Research question 3: Are assessment-based SPI programs an effective means to improve process capability for small software development firms, and is the extent of improvement associated with particular organisational characteristics?

The first question is addressed by data collected through a survey to give a broad understanding of software development practices in Queensland. Field experiments seek to answer research questions two and three by analysing reports from SPI assessments at 22 small development firms and evaluating the outcomes of the SPI program.

2.7.1 Research question 1—software best practice survey

According to diffusion of innovation theory, adoption is influenced by the characteristics of the innovation, the characteristics of the organisation adopting the innovation, and means by which the potential adopters learn about the innovation (Fichman 1992). Furthermore, the rate of adoption of an innovation is largely determined by five factors: its degree of relative advantage, compatibility, complexity, trialability and observability (Bayer & Melone 1989; Fichman & Kemerer 1993; Tornatzky & Klein 1982). Therefore, as demonstrated by Fichman and Kemerer’s (1993) comparison of adoption of structured methodologies, 4th generation languages, relational database management systems, and object oriented technologies, each software process innovation needs to be individually assessed according to the five factors. However, it is also necessary to consider community effects which are crucial for software engineering process innovations. These effects, including availability of expertise, training, and tools to support the innovation, are known in the information economics literature as network externalities (Katz & Shapiro 1985).

This research seeks to determine the extent of adoption of best practice techniques by software development organisations in Queensland. Also, in examining how widespread is the use of software process best practice techniques, the association of best practice adoption with organisational characteristics is considered.

Practice adoption levels

The ESI survey results indicated wide variation in both awareness and application of process improvement techniques (Dutta, Lee & Van Wassenhove 1999). Although many of the practices exhibited adoption rates in the range of 40-60 percent (Dutta, Kulandaiswamy & Van Wassenhove 1996), the adoption of each practice ranged from

10 percent average adoption for *analysis of test efficiency statistics* to 95 percent average adoption for *nominated project manager for each project*.

This indicates inconsistency in the use of the set of best practices included in the questionnaire: while some of the practices are widely adopted, others are rarely used. It is not surprising that some software development practices are lagging in adoption as there are barriers to adoption: some innovations are more complex and impose a knowledge burden, requiring training as well as investment in software tools (Fichman 1992). Furthermore, the perceived relevance of practices affects their adoption rate. Fitzgerald (1997) found developers are aware of methodologies and practices but 'uniquely enact' a 'methodology-in-action' for each project as deemed appropriate (p. 201).

Therefore, the first hypothesis relates to the extent of variation in the adoption of individual practices to find out if adoption of the best practices is inconsistent:

H1a There is wide variation in the adoption levels of individual best practices.

Primary lifecycle practices versus supporting and organisational practices

The second hypothesis further investigates the uneven rates of adoption of best practices. Technological process innovations relating to the core of the development activity are more observable, have higher trialability, and are perceived to be relatively more advantageous than administrative process innovations (Damanpour & Evan 1984; Swanson 1994). Furthermore, Mustonen-Ollila and Lyytinen (2003) considered four process innovation types: management; technology; tool; and description, and found that the number of innovations in each category varied over time.

The international lifecycle standard ISO 12207 identifies five primary lifecycle processes: acquisition, supply, development, operation and maintenance (ISO/IEC 2002). According to ISO 15504, the primary lifecycle processes are classified into two basic process groups: customer-supplier and engineering, and the other processes are classified as supporting lifecycle processes and organisational processes (includes management and organisation process groups).

As most software development organisations in Queensland are small firms, and small firms tend to have deep specialised technical expertise but shallow diffused management expertise (§2.6), it is expected that higher adoption of practices relating to the primary lifecycle processes would be evident, compared to adoption of supporting and organisational practices.

H1b Adoption of primary lifecycle practices is higher than adoption of supporting and organisation practices.

Wide variation in adoption by organisations

The first two hypotheses relate to the best practices, in diffusion of innovation terms, the innovation itself. The next five propositions focus on the adoption level of the organisations. As well as identifying wide variation in the adoption of individual practices, the ESI survey found the overall adoption rate by organisations varied widely (Dutta, Lee & Van Wassenhove 1999).

H1c There is wide variation in the adoption levels of best practice by organisations.

Organisation size

In previous research, as discussed in §2.5, economic, human, organisation, and implementation factors have been shown to enable or inhibit the success of SPI programs. Therefore, this study examines organisational characteristics to investigate their relationship with adoption of best practice. A number of prior studies have reported that organisational size and structural complexity associated with firm size are related to the adoption of innovations (Rogers 1995; Swanson 1994). Although organisation size could be measured by sales, revenue or assets, by far the most common metric for organisation size in Information Systems research is the number of employees (Goode 2001).

Studies of the adoption of quality assurance and certification have found that larger companies are more likely to have adopted software quality management systems than smaller companies (Burgess 1994; Smillie 1993). It may follow from this that these large companies would also be more likely than small organisations to adopt best practice. Many researchers and commentators believe small software firms are not aware of best practice, and therefore have lower adoption rates (James 2000).

As this part of the study includes in-house development groups of large organisations as well specialised software development firms, it is important to consider the size of the organisation as well as the size of the group undertaking software development. Many prior studies have not clearly made this distinction and consequently it is difficult to interpret their findings regarding the impact of organisation size on adoption of software process innovation.

In their study of small Italian software firms, Raffa and Zollo (1993) noted that small firms are disadvantaged in regard to innovation adoption. Their comments, made about innovations generally, can be applied to the adoption of software development best practice. The large company usually:

- has the financial, organisational and human resources necessary to manage a variety of activities
- can maintain software process improvement professionals inside the company
- can use effectively the past experience with methodologies to foster new process improvements
- can acquire differentiated knowledge about best practice adoption through various cooperative strategies with other firms and institutions experienced in software process improvement.

In addition, large organisations are more likely to be developing large and perhaps mission-critical systems which require best practice techniques. Findings from phase 2 of the SPICE trials suggests that larger companies have more formally defined processes compared to small organisations (Jung et al. 2001).

Therefore the following hypothesis is proposed:

H1d Adoption of software development best practice is positively associated with organisational size.

Development team size

Many researchers have found that the size of the software development group is a very significant factor in determining the kinds of practices employed, and larger software development groups tend to have more well-defined software development processes compared to smaller groups (Groves et al. 2000).

Zmud (1982) found that the size of the software development group is positively associated with technical innovation adoption (such as top-down design, structured design, structured reviews) but had a negative influence with the implementation of administrative innovations (such as chief programmer team, configuration management and unit development folder). He suggests that larger groups would have more software managers and would find it difficult to reach consensus on adoption decisions of administrative procedures. It may well be that larger groups suffer from inertia causing processes to become entrenched and resistant to change (Chiles & Choi 2000), whereas small development groups may be more innovative and flexible.

However, in analysing the relationship between quality practices and development group size in Singapore, Tan and Yap (1995) suggest that small development groups may be constrained by their resources and preoccupied with fire-fighting techniques. On the other hand, large organisations are 'able to explore greater avenues of process management, like design and code reviews, change controls and the use of sophisticated tools and techniques' (p. 234). Consequently, Tan and Yap (1995) concluded that development group size is associated with capability maturity.

Anticipating that larger organisations were better able to afford both the time and money for process improvement, Davis et al. (1993) found that the number of staff employed on software development provided a better indication of quality certification levels rather than the total number of staff in the organisation.

Therefore, the following hypothesis is proposed:

H1e Adoption level of software development best practice is positively associated with development team size.

Development type

The choice of practices employed by the organisations may also depend on their relationship with the client. Galin (2004) describes three type of software development clientele: buyers of software packages; customers of custom-made software; and internal clients. ASEI (1996) uses a similar classification for product deployment as one-off, custom or commercial-off-the-shelf (COTS). The ASEI survey reports 14 percent of employees and 40 percent of companies are producing COTS products (1996).

The closest relationship between development group and client is found with in-house software development. In this situation, the developers gain domain expertise and an understanding of the business processes, culture and politics of the parent organisation. Although such groups may gain advantages associated with large organisations, they may also be constrained by the corporate strategies and goals and therefore limited in their extent of process improvement. In Singapore, Tan and Yap (1995) noted software vendors outperformed in-house development groups in capability maturity and suggested that ‘developing software for external customers may require different levels of rigour than developing software for internal customers’ (p. 234). Software development firms undertaking custom development projects need to provide the quality demanded by the customer while balancing scope, cost and time constraints. Firms producing commercial off-the-shelf systems are often trying to define and satisfy the requirement of an unknown client. The packaged software market is dynamic, fiercely competitive and expanding on a global scale (Laitinen, Fayad & Ward 2000). Furthermore, COTS software companies have their own set of development and maintenance issues relating to the large volume of shipped product (Laitinen, Fayad & Ward 2000).

In New Zealand, Groves et al. (2000) found that more well-defined software development processes are used by teams that produce custom or COTS software rather than in-house providers. Furthermore, Jeffery and Zucker (1997) concluded that IT organisations exhibited higher adoption of metrics compared to IT user organisations.

H1f Adoption level of software development best practice varies depending on development type.

Industry sector of the development organisation

It is recognised that developers in some industry sectors are required to adhere to stringent standards, for example, safety critical systems in defence and aerospace. Furthermore, in their consideration of diffusion theory, Bayer and Melone (1989) argue that mandated software engineering innovations first introduced to a government contractor population will later transfer to the commercial sector because members of one population interact with, and in fact may jointly belong to, other populations. From the ESI surveys in Europe, Dutta, Lee and Van Wassenhove (1999) found highest adoption of software best practice by organisations belonging to the aircraft and spacecraft sector, then telecom products, and finance and insurance, followed by software consultancy and supply. In a survey of project management maturity, Ibbs and Kwak (2000) found engineering and construction sector firms exhibited the highest project management capability and IT firms were the lowest. From his experience in the software industry, Glass (1996) claims that aerospace, banking, process control, productivity tools and reservation systems exhibit higher capability compared to other application domains.

Therefore the following hypothesis is proposed:

H1g Adoption level of software development best practice varies depending on the industry sector of the development organisation.

Summary of best practice adoption hypotheses

As shown in Figure 2-2 and summarised in Table 2.5, seven hypotheses are formulated to answer the first research question. The exploration of these proposals will determine the extent of adoption of best practices by Queensland software development organisations, and the relationship between specific organisational characteristics and the extent of adoption.

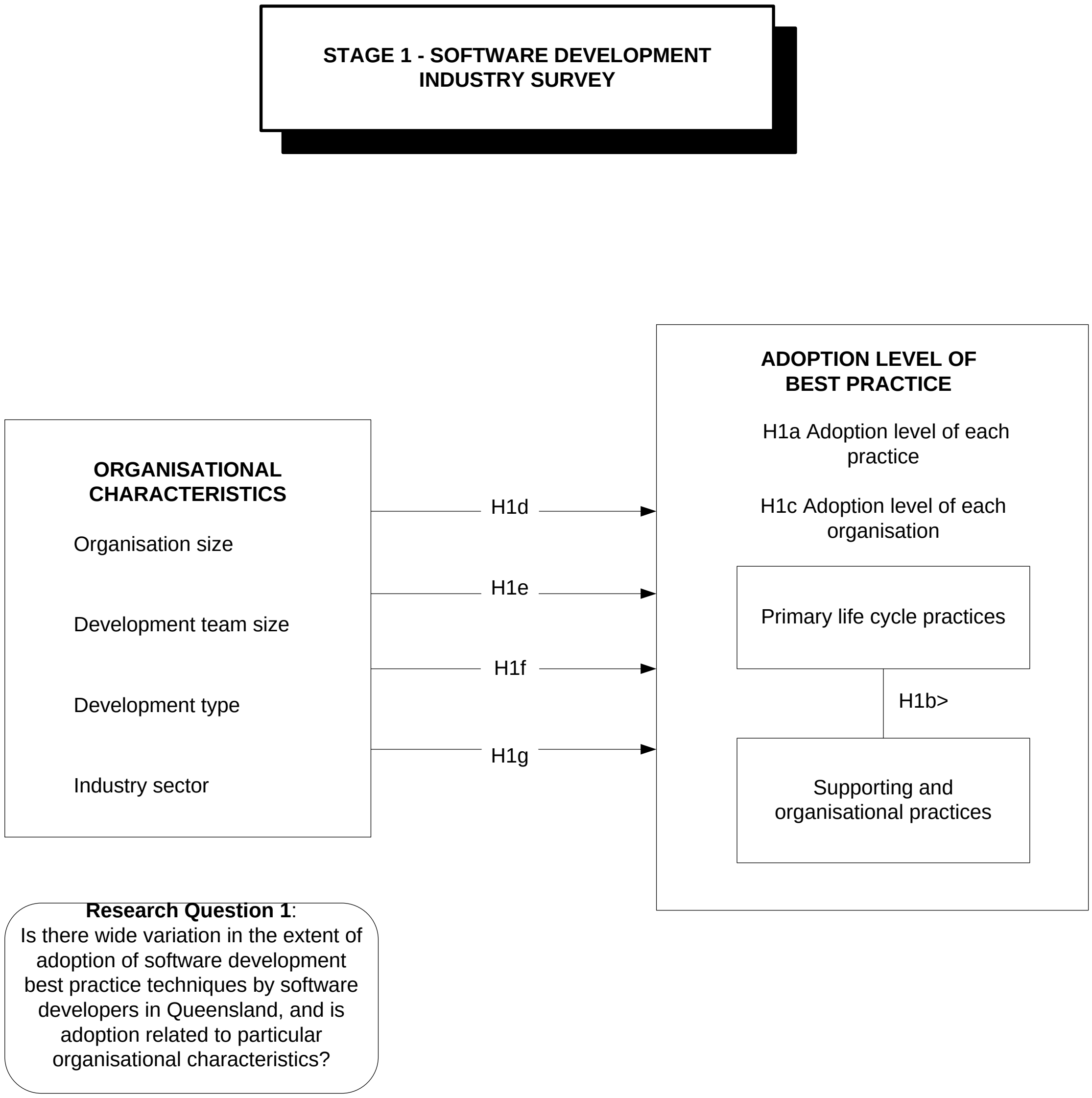


Figure 2-2 Research model showing hypotheses related to research question 1

Table 2.5 Research question 1 and associated hypotheses.

RQ1 Is there wide variation in the extent of adoption of software development best practice techniques by software developers in Queensland, and is adoption related to particular organisational characteristics?	
H1a	*There is wide variation in the adoption levels of individual best practices.*
H1b	*Adoption of primary lifecycle practices is higher than supporting and organisation practices.*
H1c	*There is wide variation in the adoption levels of best practice by organisations.*
H1d	*Adoption of software development best practice is positively associated with organisational size.*
H1e	*Adoption level of software development best practice is positively associated with development team size.*
H1f	*Adoption level of software development best practice varies depending on development type.*
H1g	*Adoption level of software development best practice varies depending on the industry sector of the development organisation.*

2.7.2 Research question 2—process capability

After exploring the level of adoption of best practice across the software development industry in Queensland, a detailed analysis of process capability in small firms is undertaken to answer the second research question: is the capability of technical processes higher than that of management processes, and is process capability associated with particular organisational characteristics?

As stated in §1.2.2, most Australian software development firms are small organisations. There has been little research to evaluate the adoption of SPI by small firms, hence this part of the study analyses the reports from an assessment-based SPI program conducted with 22 Queensland software development firms. As well as collecting assessment data, the field experiments gathered organisational and project information. In Chapter 6, a detailed description of the assessment procedures is provided. As this study is partly based on a program developed by the SQI, the range of factors is constrained by the scope and method of the assessment-based process improvement program analysed. The eight processes included in the PIP assessments are requirements elicitation, software development, configuration management, quality assurance, problem resolution, project management, risk management, and process establishment. These processes are defined and their inclusion justified in §6.5

Comparison of capability of primary lifecycle and support, management and organisation processes

In §2.7.1, based on prior research (Damanpour & Evan 1984; Mustonen-Ollila & Lyytinen 2003; Swanson 1994), it was suggested that firms would exhibit higher adoption of best practice for technical processes compared to supporting and organisational processes. As well as exploring the variation in capability of eight assessed processes, the study will test the following hypothesis:

H2a: the capability of primary lifecycle processes is higher that the capability of supporting and organisation processes.

Organisational characteristics

The next set of hypotheses consider if process capability is associated with particular organisational or project characteristics. Past studies indicate that organisational and project factors inhibit the adoption of software development methods and tools (Mustonen-Ollila & Lyytinen 2003; Prescott & Conger 1995). Diffusion of innovation theory supports the notion that firms with particular organisational characteristics would be more likely to adopt SPI, evidenced by higher process capability levels. Also, management theory provides an understanding of factors which may contribute to firms' tendency to define and formalise development processes.

Firm size

The relationship between organisation size and best practice was discussed in §2.7.1. In 27 of the 29 processes assessed in phase 2 of the SPICE trials, the average capability level for organisations with a large number of IT staff is greater than for organisations with a small IT staff, 'suggesting that larger companies have more formally defined processes' (Jung et al. 2001, p. 234). Furthermore, many researchers and consultants have claimed that formal SPI programs such as CMM or SPICE are not useful to small firms (Brodman & Johnson 1994; Coleman 2002; Kautz 1998b; Laitinen, Fayad & Ward 2000), and that higher process capability would be found in large organisations.

The findings of these studies suggest the following hypothesis:

H2b Process capability is positively associated with the number of staff employed by the firm.

Professional experience

Early research into diffusion of innovation focussed on the personal characteristics of the individual adopter rather than the organisational characteristics (Moore 1989). In researching adoption of innovation, researchers have explored the role of competence (Perry & Danzinger 1980), sophistication (Rogers 1995), and professionalism (Zmud 1984) of staff in the adopting organisation. Based on Fichman's (1995) study of software process innovation diffusion, organisations staffed by people with more extensive prior knowledge in domains of software engineering are more likely to initiate and sustain process improvement. Furthermore, Galin (2004) recommends that teams with recently hired employees 'require greater and more intense SQA efforts due to the uncertainty surrounding the members' ability to cooperate and coordinate among themselves as well as the professional uncertainty about their professional expertise and qualifications' (p. 73). The next three hypotheses relate to the knowledge and professionalism of staff employed: years of experience; formal education and technical orientation.

There are conflicting views in the research literature about the effect of professional experience on the use of formal processes. Either experienced developers will realise the benefits provided by a methodology and use it, or they may find it restraining and not use it (Fitzgerald 1997).

In studying the adoption of structured systems analysis techniques, Leonard-Barton (1987) found that more experienced analysts were more likely to use the new processes than were their less experienced colleagues. She explained this by suggesting that the veterans were more aware of the potential benefits of the new techniques. Adoption of SPI involves similar training and documentation activities to structured systems analysis, so it may be expected that experienced systems analysts would also be more likely than novices to adopt SPI. On the other hand, Fitzgerald

(1997) reports that inexperienced developers may readily adopt a methodology as it provides a useful template for the development approach.

H2c Process capability is positively associated with the proportion of experienced staff in the firm.

Formal education

In his assessment of the characteristics of adopters, Rogers found that early adopters have more years of formal education compared to late adopters (Frank et al. 2001; Rogers 1995). Furthermore, Fitzgerald (1997) suggests that educational exposure to methodologies might predispose developers towards a more formalised methodological approach.

H2d Process capability is positively associated with the proportion of formally qualified staff in the firm.

Proportion of technical staff

In formulating hypothesis H1e in §2.7.1, it was noted that the size of the software development group is a very significant factor in determining the kinds of practices employed. However, the size of the firm in terms of the number of employees does not provide accurate information about the number of people involved in software development. In most software development firms in Queensland, staff are employed under either a technical or administrative employment award (QPS 2003). The technical award covers programmers, analysts and testers, whereas the administrative award relates to clerical and support staff and managers. Rogers (1995) found that professional orientation is associated with adoption of innovation. Therefore, it is expected that firms with a higher proportion of technical staff would have a more professional orientation and would exhibit higher process capability levels compared to firms with a lower proportion of technical staff.

H2e Process capability is positively associated with the proportion of technical staff in the firm.

Years of operation

Eisenhardt (1988) suggests that organisations adopt processes and practices common at their time of founding, and that these policies are resistant to change, even in the face of major changes in job content and technology. This suggests that with the recent increase in SPI awareness and adoption (Sweeney Research 2003a), recently founded organisations would be more likely to implement SPI than longer established ones. In a Korean study of large in-house development groups, Lee and Kim (1992) studied the formalisation of system development procedures and expected that the degree of procedural formalisation would be positively associated with the age of the MIS department. They were surprised to find a significant negative outcome, and acknowledge the inertia effect which may prevent the adoption of modern software development methodologies by older MIS departments. However, the findings from McCaffery et al. (2004) from Northern Ireland disagree: older, more established government and telecom organisations make best use of formal methodologies.

Startup companies have limited resources and therefore find it difficult to balance the enforcement of processes while maintaining flexibility and struggling to survive (Ho & McGuire 1998; Laitinen, Fayad & Ward 2000). While recognising that process definition can offer advantages to small recently founded firms, Coleman (2002) warns that in the case of the software start-up firm, where no history of software development exists, using the correct practices can be the difference between survival and demise. Evidence suggests that many small software firms do not stay in business for more than five years, they get absorbed by other firms or cease to operate (Raffa, Zollo & Caponi 1996).

This suggests the following hypothesis:

H2f Process capability is positively associated with the length of time the firm has been in operation.

Target business sector

As discussed §2.7.1, research suggests that the industry sector of the software development organisation influences the adoption of software development innovations (Bayer & Melone 1989; Dutta, Lee & Van Wassenhove 1999; Glass

1996). This is particularly relevant for in-house development groups, but most small development firms belong to the IT sector. However, to be successful, software development firms need to be aware of the standards required by their clients. Therefore, the requirements of client organisations in the business sector, targeted as the market for the software, affects the processes adopted by the software development firms. Any firm supplying safety critical systems will undergo second or third party certification to ensure relevant standards are met. Furthermore, large purchasers such as government departments, public utilities and telecommunications firms have gone through phases of enforcing and relaxing mandatory standards (Black 1996; Gome 1996a; Gottliebsen 1995; Hilvert 1996; Inwood 1993; Liackman 1994; Sharman 1992).

Although phase 2 of the SPICE trials collected information relating to participants' primary business sector and target business sector, an analysis of the relationship of capability levels with either measure of industry sector has not been reported (Hunter 1998). The following hypothesis is suggested:

H2g Process capability is associated with specific target business sectors.

ISO 9000 certification

If a software organisation uses ISO 9000 certification to design its quality system, benefits can be achieved in terms of meeting the requirements of the customers (Jenner 1995), and measuring conformance to requirements (Crosby 1979). As noted by Jung et al. (2001), ISO 9001 and ISO 15504 have different origins but are intuitively similar. In an extensive mapping exercise, Hailey (1998) found that all 200 base practices of ISO/IEC PDTR 15504 corresponded to at least one of the 20 clauses of ISO 9001. The SPICE phase 2 trials reported that the average capability level for many of the assessed processes of ISO 9001 certified organisations is at least level 2. Furthermore, for ten of the assessed processes, the capability level of the ISO 9001 certified organisations is significantly greater than that of non-certified organisations (Jung et al. 2001).

In a short period of time, companies in India have successfully embraced SPI adoption with many firms achieving high CMM levels (Mohnot 1995; Phillips 2003). Adoption of ISO 9000 has been attributed as the single biggest factor for Indian companies moving to CMM, as the early adoption of ISO 9000 brought in a process orientation and facilitated the adoption of CMM (Jalote 2001). Therefore, the following hypothesis is suggested:

H2h Process capability is positively associated with certification to ISO 9000.

Project characteristics

Although organisation characteristics have attracted the attention of researchers, little research has focussed on characteristics of projects undertaken by firms. Beecham, Hall and Rainer concluded from their study of SPI problems in 12 UK companies, that organisational issues are more of a problem for high maturity firms while 'low maturity companies suffer from project and technical problems' (2001, p. 12). The four project characteristics examined in this study are the number of projects in progress at each firm, the typical size of project teams, the typical duration of each project, and the extent of project cost overrun. These four factors are interrelated due to the trade off between effort and duration.

The risk of failure in an IT project is a factor of three elements: the project size, and the organisation's experience in performing projects of this size; the developers' experience with the technology; and finally, project structure which incorporates task uncertainty and organisational change requirements (Applegate, Austin & McFarlan 2003, pp. 580-1). Increasing the size of IT projects greatly increases their technical complexity. Brooks (1975) reported the effort to write a piece of code grows at an exponent of 1.5 of the number of lines of code. Increased risk is also reflected in the costs of communicating in a large team, with the number of communication channels in the team equal to $n(n-1)/2$ where n represents the number of people in the team. In addition, the greater the number of tasks required to complete a project, the greater the probability of failure, simply because there are more tasks that can fail and the outcome is dependent on successful completion of all tasks (Thorogood & Yetton 2004).

Number of projects in progress

As a software firm takes on more projects, the need to coordinate and control the work requires a higher degree of formalisation of procedures. Each project will need to be managed, imposing a degree of structural hierarchy. Rogers (1995) found that more highly structured organisations are early adopters of innovations. The number of projects in progress could also be a used as a measure of organisation size, in the same way that hospitals use number of beds as an organisation size metric (Goode 2001). Therefore, the following hypothesis is suggested:

H2i Process capability is positively associated with the number of projects in progress.

Size of the project team

For the reasons mentioned above, the larger the project team, the more necessary it would be to have defined processes to facilitate communication and coordination. The size of the project team will also impact on the project duration. The non-linear relationship between effort and duration was famously stated in the 'Mythical Man Month' in 1975 as Brooks' Law: adding manpower to a late software project makes it later (Brooks 1975).

H2j Process capability is positively associated with size of the project team.

Project time duration

Organisations are under increasing pressure to complete projects to meet market deadlines. Methods such as rapid application development, object-oriented development, and agile methods all aim to reduce the duration of project development (Fayad, Laitinen & Ward 2000). There are conflicting arguments about the relationship between project duration and process capability levels. On one hand, it is claimed (SPIRE 1998) that higher process capability provides shorter time to market, therefore firms with higher process capability will have reduced the duration of projects. On the other hand, a longer project may be more complex and more critical, and therefore need formalised processes of higher capability compared to a shorter project. Furthermore, the longer project duration runs the risk of increased staff turnover, therefore requiring formalised processes to avoid schedule disruption.

Consequently, although recognising that higher levels of process capability should reduce the uncertainty of project duration estimates, and help meet project deadlines, it is expected that projects of longer durations would require more formalised processes.

H2k Process capability is positively associated with the typical time duration of the project.

Project cost overrun

Organisations with processes at low levels of capability are urged to adopt SPI to reduce costs and defects. Lawlis, Flowe and Thordahl (1995) reported improved cost and schedule performance with increases in process maturity. Higher process capability promises improved estimating procedures, tighter project control, improved defect detection and removal. All these improvements should reduce the typical project cost overrun.

H2l Process capability is negatively associated with the typical cost overrun of projects.

Perceptions of firm performance

As detailed in §2.2.1, empirical research shows that TQM in general and process improvement in particular can have a positive impact on the performance of the firm (Adam 1994; Anderson & Sohal 1998; Dale et al. 2001; U.S. General Accounting Office 1991; Zhang 2000). From the literature, it has been shown that assessment-based SPI initiatives, by articulating and institutionalising 'best-practice' processes should help the firm to meet goals of cost, schedule, functionality and quality. Furthermore, studies of benefits from the implementation of assessment-based SPI models such as CMM and SPICE have reported financial returns, reduction in defects, and improvements in productivity, quality and schedule (Batista & Dias de Figueiredo 2000; Brodman & Johnson 1995; Krasner 1997; Lawlis, Flowe & Thordahl 1995; McGibbon 1999; Solon & Statz 2002; Yamamura & Wigle 1997).

However, it is difficult to gather objective performance measures, especially in small firms. Many firms do not record performance metrics relating to budget, schedule, productivity, and others prefer to keep such measures confidential. From a large survey of organisations which have implemented CMM, Goldenson and Herbsleb (1995) evaluated the benefits of higher process capability using six subjective measures of performance: ability to meet schedule, budget, product quality, staff productivity, staff morale, and customer satisfaction. Their analysis found that organisations with higher capability tend to perform better on the following dimensions: ability to meet schedule, product quality, staff productivity, customer satisfaction, and staff morale. The relationship with the ability to meet budget commitments was not found to be statistically significant, but all other relationships were statistically significant. Furthermore, their analysis revealed that size did not matter: ‘organisations with relatively few software employees appear to benefit from higher process maturity just as do larger organisations’ (Goldenson & Herbsleb 1995, p. 7).

However, a different conclusion was reached from the SPICE phase 2 trials. El Emam and Birk (2000a) found that for small organisations (less than 50 staff), schedule performance is associated with the achievement of process attributes for the *develop software design* process. For large organisations (more than 50 staff), multiple associations were reported: budget performance is associated with attribute achievement of *develop software design* and *implement software design*; schedule, customer, requirements and staff morale performance are all associated with *develop software design*; and finally, productivity performance is associated with both *develop software requirements* and *integrate and test system* processes. Therefore, the following hypothesis is suggested:

H2m Process capability is positively associated with perceptions of higher firm performance.

Summary of process capability hypotheses

As shown in Figure 2-3 and summarised in Table 2.6, 13 hypotheses are formulated to answer the second research question. The exploration of these propositions will provide an understanding of the process capability of small software development firms and the relationship between specific organisational characteristics and process capability.

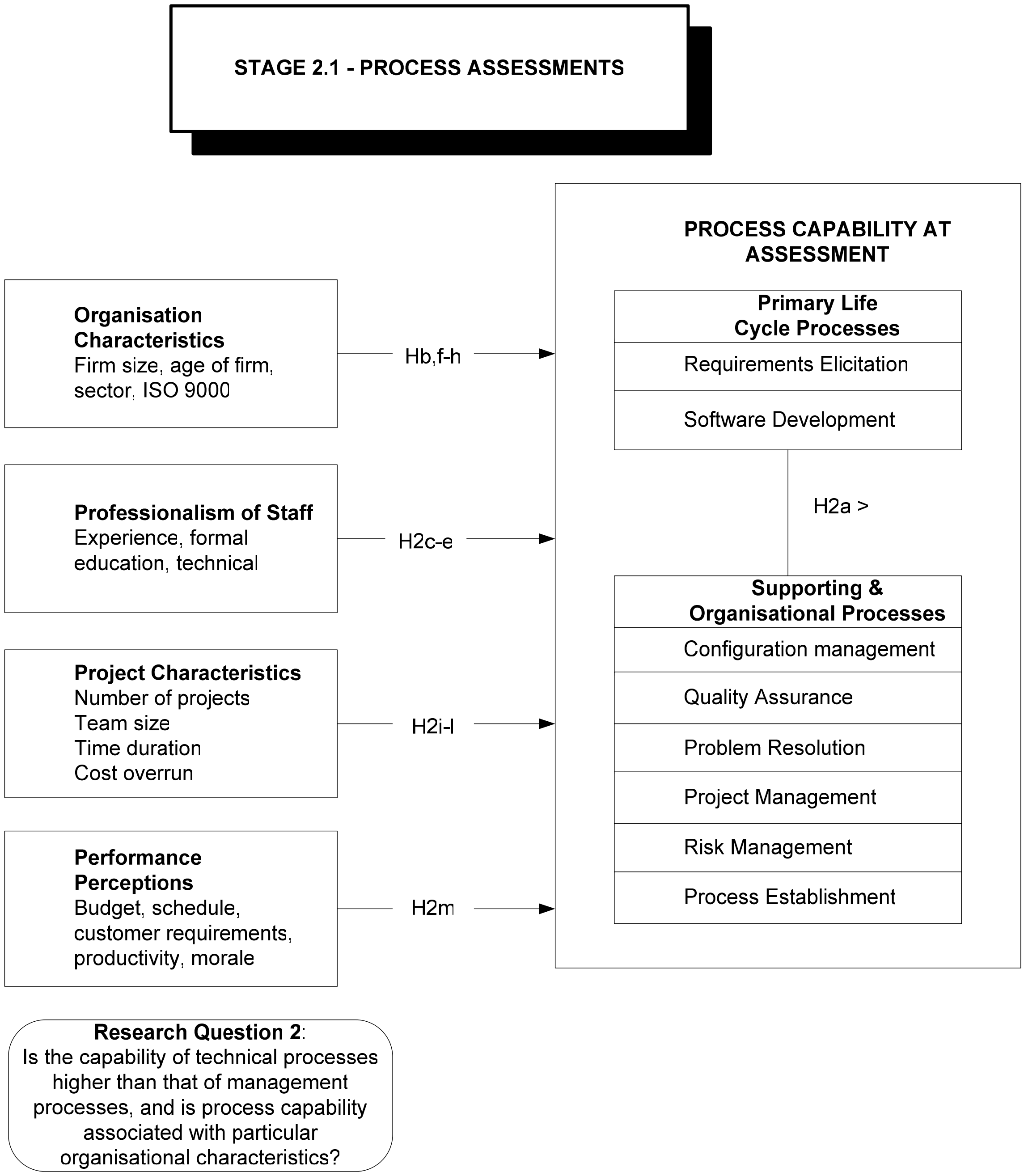


Figure 2-3 Research model showing hypotheses related to research question 2

Table 2.6 Research question 2 and associated hypotheses.

RQ2 Is the capability of technical processes higher than that of management processes, and is process capability associated with particular organisational characteristics?	
H2a	*The capability of primary lifecycle processes is higher than the capability of supporting and organisation processes.*
H2b	*Process capability is positively associated with the number of staff employed by the firm.*
H2c	*Process capability is positively associated with the proportion of experienced staff in the firm.*
H2d	*Process capability is positively associated with the proportion of formally qualified staff in the firm.*
H2e	*Process capability is positively associated with the proportion of technical staff in the firm.*
H2f	*Process capability is positively associated with the length of time the firm has been in operation.*
H2g	*Process capability is positively associated with target business sectors.*
H2h	*Process capability is positively associated with certification to ISO 9000.*
H2i	*Process capability is positively associated with the number of projects in progress.*
H2j	*Process capability is positively associated with the size of the project team.*
H2k	*Process capability is positively associated with the typical time duration of the project.*
H2l	*Process capability is negatively associated with the typical cost overrun of projects.*
H2m	*Process capability is positively associated with perceptions of higher firm performance.*

2.7.3 Research question 3—issues related to extent of process improvement

Although there are many reports about successful SPI adoption and the resultant benefits, research also shows that many assessment-based SPI initiatives are not successful, inhibited by many of the barriers identified in §2.5. In the late 1980s, two thirds of all SPI programs faltered or failed after the initial assessment due to flawed strategy, lack of commitment, lack of follow-through, not measuring improvements, and lack of crisp SPI objectives tied to business objectives (Krasner 1997). Even organisations that proceed with the SW-CMM program to a reassessment do not all exhibit increases in maturity levels. In comparing the first and latest reassessments of 314 organisations, the SEI found that 15 percent had no change in their capability level

(Software Engineering Institute 2000). Furthermore, the results of a survey of 14 SPICE phase 1 trials organisations 'indicate that many organisations struggle with achieving successful SPI based on process assessments' (SPICE Project 1998, p. 164).

The third research question addressed in this study asks if assessment-based SPI programs are an effective means to improve the process capability of small software development firms, and is the extent of improvement related to particular organisation characteristics? To date, there has been very little empirical research relating to reassessments of process capability (Goldenson & Herbsleb 1995). As mentioned previously, the SEI provides a regular summary report on the number of CMM reassessments and the extent of improvement in capability levels, but this information is not reported with the time period from first to latest assessment, or linked to organisational characteristics such as size (Software Engineering Institute 2000).

In a survey of 56 CMM assessed organisations, Goldenson and Herbsleb (1995) asked managers to estimate their current maturity levels one to three years after their assessment. There was little difference reported, with most organisations stating their capability level had not improved (77% still at level 1; 79% remained at level 2). A more positive outcome resulted from a software process improvement program in Finland: 'the companies have been able to increase their process capability: typically one level in two years for the selected processes' (Varkoi 2004, p. 8).

Increase in process capability level

According to the SPICE Project team, 'assessments are considered effective if their findings determine an organisation's SPI effort, and if the organisation is successful in addressing the findings of the assessment' (SPICE Project 1998, p. 163). The process improvement program reported in this thesis included a return visit of an assessor to each firm. This follow-up meeting afforded the opportunity to determine the extent of improvement in terms of process improvement. It was hypothesised that:

H3a: Process capability is higher at the follow-up meeting compared to process capability at the initial assessment.

Firm size and extent of improvement

It was anticipated that the extent of improvement would vary across the firms involved in the SPI program. Recognising that organisational characteristics affect the adoption of innovations, it was anticipated that the size of the firm may affect the extent to which the assessment recommendations could be implemented. Defining and developing processes can be resource intensive and may need specialist support, therefore, it is expected that larger firms would be better placed to achieve such change.

H3b: The extent of improvement is positively associated with the size of the firm.

Staff professionalism and extent of improvement

As discussed in §2.7.2, the professionalism of staff involved in adopting the innovation is another factor linked to higher adoption rates. As professionalism includes the professional experience of staff as well as their formal education qualifications, it is expected that these factors would be associated with successful implementation of the assessment recommendations.

H3c: The extent of improvement is positively associated with the proportion of experienced staff.

H3d: The extent of improvement is positively associated with the proportion of formally educated staff.

H3e: The extent of improvement is positively associated with the proportion of technical staff.

Summary of assessment evaluation hypotheses

As shown in Figure 2-4 and summarised in Table 2.7, five hypotheses are formulated to answer the third research question. The exploration of these propositions will provide an understanding of the effectiveness of the process improvement project for small software development firms, and the relationship between specific organisational characteristics and process capability.

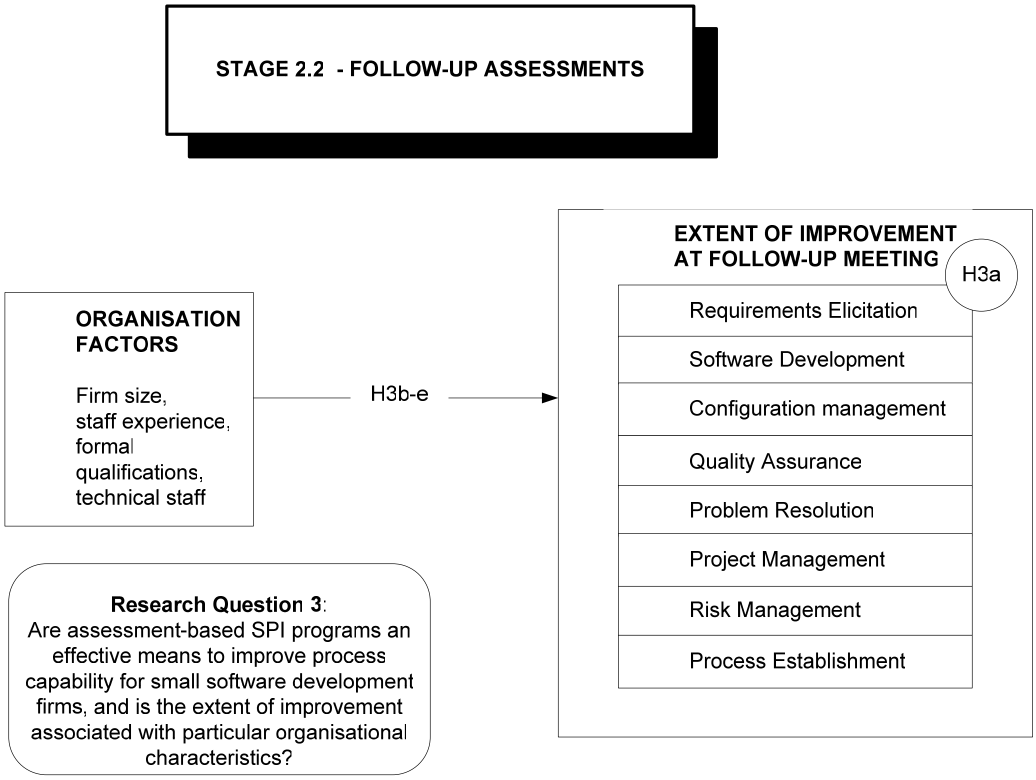


Figure 2-4 Research model showing hypotheses related to research question 3

Table 2.7 Research question 3 and associated hypotheses.

RQ3 Are assessment-based SPI programs an effective means to improve the process capability of small software development firms, and is the extent of improvement related to particular organisational characteristics?	
H3a	*Process capability at the follow-up meeting is higher compared to the initial assessment.*
H3b	*The extent of improvement in process capability is positively associated with the size of the firm.*
H3c	*The extent of improvement in process capability is positively associated with the proportion of experienced staff.*
H3d	*The extent of improvement in process capability is positively associated with the proportion of formally educated staff.*
H3e	*The extent of improvement in process capability is positively associated with the proportion of technical staff.*

2.8 Conclusion

This chapter has provided the underlying theories of process improvement and explained how process improvement is an integral part of TQM. To understand the adoption and diffusion of process innovations, diffusion of innovation theory was introduced to provide a theoretical framework. Moving from management and organisational research to a software-specific focus, two relevant SPI models were introduced: CMM and SPICE. Empirical research reporting the adoption of best practice was then presented, followed by a summary of surveys, case studies and field experiments related to SPI. This summary was structured to illustrate the diverse range of issues identified as critical success factors for SPI adoption. This review of research to date highlighted the pressing need to consider the special issues faced by small software development firms undertaking SPI. Based on the literature reviewed, the research model was presented, and the research questions introduced in Chapter 1 were then extended into specific, testable hypotheses.

The next chapter will describe and justify the overall research approach.

3 CHAPTER THREE—RESEARCH APPROACH

3.1 Introduction

In the previous chapter, literature relating to the underlying theories of process improvement and diffusion of innovation was reviewed and then current research about software process best practice and improvement was summarised. From the literature review, a gap was identified: more research is needed about improving software processes in small software development firms.

The aim of this chapter is to describe the research design. The research design enables collection of data to investigate the hypotheses formulated in Chapter 2. The hypotheses were developed from research questions raised in Chapter 1. The function of this chapter is to explain the research paradigm, describe the approach and introduce the methods used to collect and analyse the empirical measures. Also, the adopted research approach is justified.

The first section provides a brief account of research approaches traditionally used in the software engineering field. The underlying paradigm of this research is discussed in §3.3 and the research approach adopted in §3.4. An overview of the two research methods is provided in §3.5. In §3.6, reference disciplines for the methodology are discussed. The unit of analysis is explained in §3.7, and the final section summarises the research plan. The survey instrument and procedures are described in Chapter 4, and the field experiments in Chapter 6.

3.2 Software engineering research approaches

This first section seeks to answer the question: what approaches are traditionally taken in software engineering (SE) research, with software engineering defined as: ‘(1) the application of systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software; (2) the study of approaches as in (1)’ (IEEE 1990).

Glass, Vessey and Ramesh (2002) analysed 369 papers covering the time period from 1996 to 1999 to discover the essence of SE research. The research papers were randomly sampled from six key journals representing software engineering literature:

Information and Software Technology; Journal of Systems and Software; Software Practice and Experience; IEEE Software; ACM Transactions on Software Engineering and Methodology; and *IEEE Transactions on Software Engineering*. In the analysis of these papers, the research approach was categorised at the highest level as descriptive, evaluative or formulative, and 22 different research method categories were identified. Glass, Vessey and Ramesh concluded that SE research is *diverse* regarding topic, *narrow* regarding research approach (the *formulative* category represented 55% of the sample), *narrow* in terms of method (conceptual analysis 43%), *inwardly-focused* regarding reference discipline (98% used no reference discipline), and *technically focused*, as opposed to behaviourally focused, regarding unit of analysis (Glass, Vessey & Ramesh 2002).

It is recognised that the computing research area is made up of three major disciplines: (management) information systems, computer science and software engineering, and that a degree of overlap exists between these three fields (Glass, Vessey & Ramesh 2002). Over the years, a number of writers have identified software engineering and engineering-type research as a legitimate part of management information systems (MIS) research (Culnan & Swanson 1986; Davis 1983; Morrison & George 1995; Vogel & Wetherbe 1984). More recent examples of software engineering in the MIS research area include Cecez-Kecmanovic (1994) in relation to engineering-type research; Iivari, Hirschheim and Klein's (1998) constructive-type research; Baskerville and Woodharper's (1998) prototyping; and both Lau (1997), and Burstein and Gregor's (1999) systems development approach.

In order to explore the software engineering component in MIS research, Morrison and George (1995) examined 1,180 research articles published in three leading MIS related journals: *Communications of the ACM* (393 articles); *Management Science* (614 articles); and *MIS Quarterly* (173 articles) over a six year period to December 1991. They distinguished the overlapping or combined domain of MIS and software engineering research from other types of MIS or software engineering research as follows: 'it involves all aspects of IS development from an organisational perspective. In other words, as systems development where organisational factors (e.g., processes, requirements, opinions, etc.) are a fundamental component of the research questions and goals' (p. 81). From this analysis, they determined that almost 45 percent of all

MIS-related articles belonged to the combined domain of MIS and SE (MIS/SE). According to their definition, the research reported in this thesis would fit into their MIS/SE research as it focuses on organisational factors as well as technical and management processes.

The need to include the organisational context in software engineering research was also acknowledged recently by other researchers, such as Edwards and Thompson (2003), who 'believe that research into the application and management of software engineering is of fundamental importance within the software engineering field and that this aspect is often overlooked in favour of a focus on technical solutions' .

Therefore, in this study, research approaches and methods commonly used in the IS discipline (and borrowed from other disciplines) are used, but it is recognised that the problem domain lies mainly in the software engineering research field.

3.3 Research philosophy

Before selecting an approach and methods for the research, the underlying paradigm is considered. Paradigm is defined as 'the basic belief system or worldview that guides the investigator, not only in choices of method but in ontologically and epistemologically fundamental ways' (Guba & Lincoln 1994, p. 105). The concepts of epistemology and ontology underpin the selection of the research approach (Cornford & Smithson 1996; Falconer & Mackay 1999; Guba & Lincoln 1994), and are now discussed in relation to this study.

Epistemology refers to the type of knowledge that can be obtained about the phenomena being studied. On one hand, a positivist epistemology sees the researcher as an objective neutral observer seeking regularities and causal relationships by firstly formulating research questions and hypotheses, then specifying research strategies and analysis methods (Falconer & Mackay 1999). At the other end of the epistemological continuum are the non-positivists (Falconer & Mackay 1999), or anti-positivists (Cornford & Smithson 1996), who see the researcher in a subjective role, directly involved in the activity, and tending to gather data prior to committing to theoretical constructs or hypotheses (Falconer & Mackay 1999; Kaplan & Duchon 1988). Some writers break the non-positivist epistemology down into the interpretive and the critical paradigms (Myers 1997; Orlikowski & Baroudi 1991). The interpretive approach focuses on the context of the situation and attempts to understand phenomena through the sense-making or meanings that people assign to them (Myers 1997). The critical research approach recognises that although people can choose to change their social and economic circumstances, they are constrained by social, cultural and political forces (Myers 1997).

Ontology. The degree of subjectivity versus objectivity also lies at the heart of the concept of ontology, but in terms of reality as opposed to knowledge. Nominalism or idealism espouses an individual concept of reality, whereas realists believe things exist in a real and concrete world (Cornford & Smithson 1996), and that it is possible to know '*how things really are* and *how things really work*' (Guba & Lincoln 1994, p. 108). Methods selected for this research are consistent with an objective view, and it is assumed that a real world exists to be objective about.

For this study, the *positivist* paradigm is employed to determine the extent of adoption of best practice techniques by software development organisations, and to explore and evaluate the impact of the software process improvement program on a group of small software development firms. Both the SE and IS disciplines demonstrate a strong positivist tradition. It has been reported that almost all IS research in the United States and Australia conformed to a positivist epistemology (Alavi & Carlson 1992; Falconer & Mackay 1999; Orlikowski & Baroudi 1991; Ridley & Keen 1999). It appears that Australia follows North America in this regard, as a comparison study of IS dissertations highlighted North American IS dissertations were predominantly quantitative and empirical, whereas dissertations from Europe favoured a qualitative non-empirical approach (Evaristo & Karahanna 1997).

3.4 Research approach

There are many different taxonomies used to define the research approach. Adopting the terminology from Galliers (1991), research *approach* describes the general style of research endeavour, and a *method* is the application of a set of distinct techniques. Approaches 'embody a particular style and many employ different methods' (Galliers 1991, p. 329). The four generally-accepted research approaches reported by Morrison and George (1995) are formulative, evaluative, descriptive, and developmental. Using this classification scheme, this study can be defined as using an *evaluative* research approach as it involves a methodology based on a scientific method, and generates and tests hypotheses. The extent of adoption of best practice by software development organisations is evaluated, as is the SPI experience of a selected group of organisations. There is also an element of descriptive research as the practices used by developers are described.

A different classification of research styles is suggested by Cornford and Smithson (1996) who extended Iivari's work (1991). In this framework of three categories of conception of the research task, the *constructive* style includes conceptual and technical development; *nomothetic* style includes formal-mathematical analysis, laboratory and field experiments, field studies and surveys; and the *idiographic* style represents case studies and action research (Cornford & Smithson 1996). This study conforms to the *nomothetic* style as it uses a survey and field experiments to test hypotheses within the scientific tradition. It has been noted that US academics, peer reviewers and publishers tend to favour a nomothetic approach (general laws and procedures of exact science) in contrast to Europeans academics who prefer an ideographic one (an understanding of particular cases) (Bengtsson, Elg & Lind 1997).

According to Cornford and Smithson (1996), a popular classification, dubbed the *classic* taxonomy of styles of research within IS research was described by Vogel and Wetherbe (1984) and included case studies, field experiments, field studies, laboratory experiments, conceptual studies, reviews and tutorials (Vogel & Wetherbe 1984).

This classic taxonomy was further developed by Galliers (1991) who classified IS research approaches into two broad streams: *scientific* (empirical) and *interpretivist*. The scientific approach is based on the basic assumptions of the positivist paradigm: repeatability, reductionism and refutability. The approach taken in this study is consistent with the scientific approach as it uses prior theory to determine the observations needed to test the theory. On the other hand, applying the interpretivist approach would see the world of SE as 'something that can only be interpreted, never fully specified or reduced to theories' (Cornford & Smithson 1996, p. 47). In the scientific stream, Galliers (1991) includes laboratory experiments, field experiments, surveys, case studies, theorem proof, forecasting and simulation; and in the interpretivist stream: subjective/argumentative, reviews, action research, descriptive/interpretive, futures research and role/game playing (Galliers 1991, p. 332). One would expect Gallier's scientific/interpretivist split to match Evaristo and Karahanna's (1997) empirical/non-empirical classification, however, Evaristo and Karahanna classify case studies (including action research and participant observation) as empirical research methods. The lack of consistency in definition of approaches and methods poses a challenge to researchers when classifying and comparing SE research.

3.4.1 Theoretical and empirical aspects

Another aspect of the research approach is the extent to which the study can be classified as *theoretical* or *empirical*. In the past, there has been a significant body of theoretical research in the SE field characterised by researchers applying a 'mental set of procedures' to 'develop and refine a body of abstract understanding of phenomena and issues' (Cornford & Smithson 1996, p. 43). On the other hand, empirical research is based on the 'observation of actual practice for the purpose of discovering the unknown or testing a hypothesis' (Brilliant & Knight 1999). Methods used in empirical research include gathering data by experiment, observation or demonstration of technology (Brilliant & Knight 1999). Therefore, empirical research is based on observed and measured phenomena and derives knowledge from actual experience rather than from theory or belief (Grogg). According to Perry, Porter and Votta (2000), the essence of an empirical study is the attempt to learn something useful by comparing theory to reality, thereby enabling theories to be improved. The study reported in this thesis follows the steps for an empirical study (as recommended by Perry, Porter & Votta 2000): formulate an hypothesis or question to test; observe a situation; abstract observations into data; analyse the data; and finally, draw conclusions with respect to the tested hypotheses.

3.4.2 Empirical approach in software engineering research

Since 1993, heightened awareness of the need for empirical studies in the SE field has resulted in an increase in empirical SE research, as evidenced by journals (such as the *Empirical Software Engineering Journal*), research centres (CeBASE NSF Center for Empirically Based Software Engineering, University of Maryland and Fraunhofer Institute for Experimental Software Engineering; Centre for Empirical Software

Process Research (CESPR) University of Hertfordshire, UK) and conferences (International Symposium on Empirical Software Engineering) focusing on this type of approach (Shull, Carver & Travassos 2001). It has been recognised that, prior to this time, there existed a serious dearth of empirical studies and evidence (Basili 1993; Brilliant & Knight 1999; El Emam & Goldenson 1998; Fenton 1993b; Fenton, Pfleeger & Glass 1994; Perry, Porter & Votta 2000; Tichy et al. 1995; Xia 1998; Zelkowitz & Wallace 1997). The catalyst to focus attention on an empirical approach in SE research was an international workshop on experimental software engineering issues which was held at Dagstuhl Castle in Germany in 1993. It is claimed that this workshop strengthened empirical software engineering as a major sub-discipline of software engineering (Jeffery & Votta 1999). However, today there are still claims of a chasm between theory and practice, and researchers are urged to conduct evaluative research rather than advocatory research (Glass 2003a, pp. 478-9).

Empirical studies are considered very important to the software engineering field as the results of empirical research help to characterise SE technical problems and provide the means to evaluate new techniques in the relevant context (Basili 1996; Brilliant & Knight 1999). However, some researchers claim that much experimental software engineering research lacks rigour (Fenton, Pfleeger & Glass 1994; Perry, Porter & Votta 2000; Shaw 2003) and these writers urge researchers to create better research designs and to validate the studies undertaken. Furthermore, field studies are considered to be better than laboratory-style experiments due to concerns with feasibility and scalability of laboratory-style experiments 'because we are not interested in toy problems!' (Agresti 1993a), and as expressed by Fitzgerald (1997), it is better to study the elephant in the jungle, not in the zoo.

It is recognised that a series of randomised experiments would be worthwhile to investigate the influence of organisational characteristics such as size and ISO 9000 certification, however, such randomised experiments are not possible due to cost, ethics or legal reasons (Jung et al. 2001, p. 234).

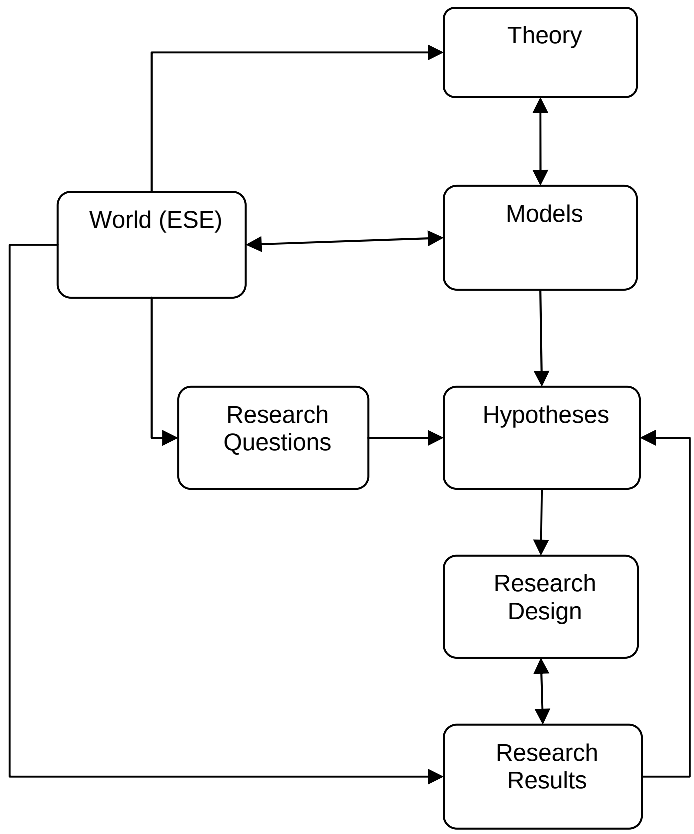


Figure 3-1 Empirical Software Engineering research approach
(Jeffery & Votta 1999)

This study conforms to the empirical software engineering (ESE) research model promoted by Jeffery and Votta (1999, p. 436) and has been adapted and presented here as Figure 3-1.

3.5 Research method

The previous sections detailed the selection of the research paradigm and approach. In this section an introduction is provided to the research method which incorporates two complementary methods: survey and field experiments.

3.5.1 Multi-method approach

Traditionally, researchers in the computing field have tended not to use multiple research methods to collect data. Recently, however, this view has been challenged, as it has been recognised that applying a variety of methods can provide a richer, contextual basis for interpreting and validating results (Brewer & Hunter 1989; Gable 1994; Gallivan 1997; Jick 1979; Kaplan & Duchon 1988; Lee 1991; Mingers 2001; Sawyer 2001; Seaman 1999; Singer, Storey & Sim 2000; Wood et al. 1999). The benefits of a multi-method approach are promoted by Morrison and George:

> 'By recognising and using the broad spectrum of the MIS/SE research domain, practical and beneficial research can result. Such research could address software-related problems currently confronting organisations. An interdisciplinary, multi-dimensional, multi-faceted approach to research involving software development…provides a vehicle for mutually beneficial research among both organisationally-oriented and technically-oriented researchers' (1995, p. 90).

Furthermore, software engineering researchers are urged to use quantitative analysis focusing on statistical analysis of numerical data, as well as qualitative analysis focusing on textual and numerical data (Carver 2003; Scacchi 1993). The combination of qualitative and quantitative methods is promoted as providing the advantages of triangulation and synergy (Iversen et al. 1998, p. 464).

It has been noted that survey and fieldwork approaches are complementary for information technology research, as traditional survey work is strong in areas where field methods are weak (Attewell & Rule 1991; Gable 1994). The use of survey and in-depth interviews enables both a broad view of the industry as a whole, and a more detailed picture of a few companies (Groves et al. 2000).

This research meets the criteria of mixed methods research defined by Gallivan (1997) as it is empirical research using at least two different methods for collecting data; one of the data collection methods is quantitative (survey) and the other is qualitative (process assessments); both qualitative and quantitative data are presented and analysed; and the researcher addresses a theoretical question rather than providing description only. A superior piece of research can be achieved by combining the main strength of survey research, generalizability (external validity), with the main strength of experimentation (internal validity) (Gutek 1991). The mixed method approach used in this study provides richer data and the opportunity to compare the best practice survey results with the results from the process improvement program. Figure 3-2 was included in Chapter 1 to graphically represent the structure of the thesis. This figure is repeated here to illustrate the integration of the survey and field experiments methods.

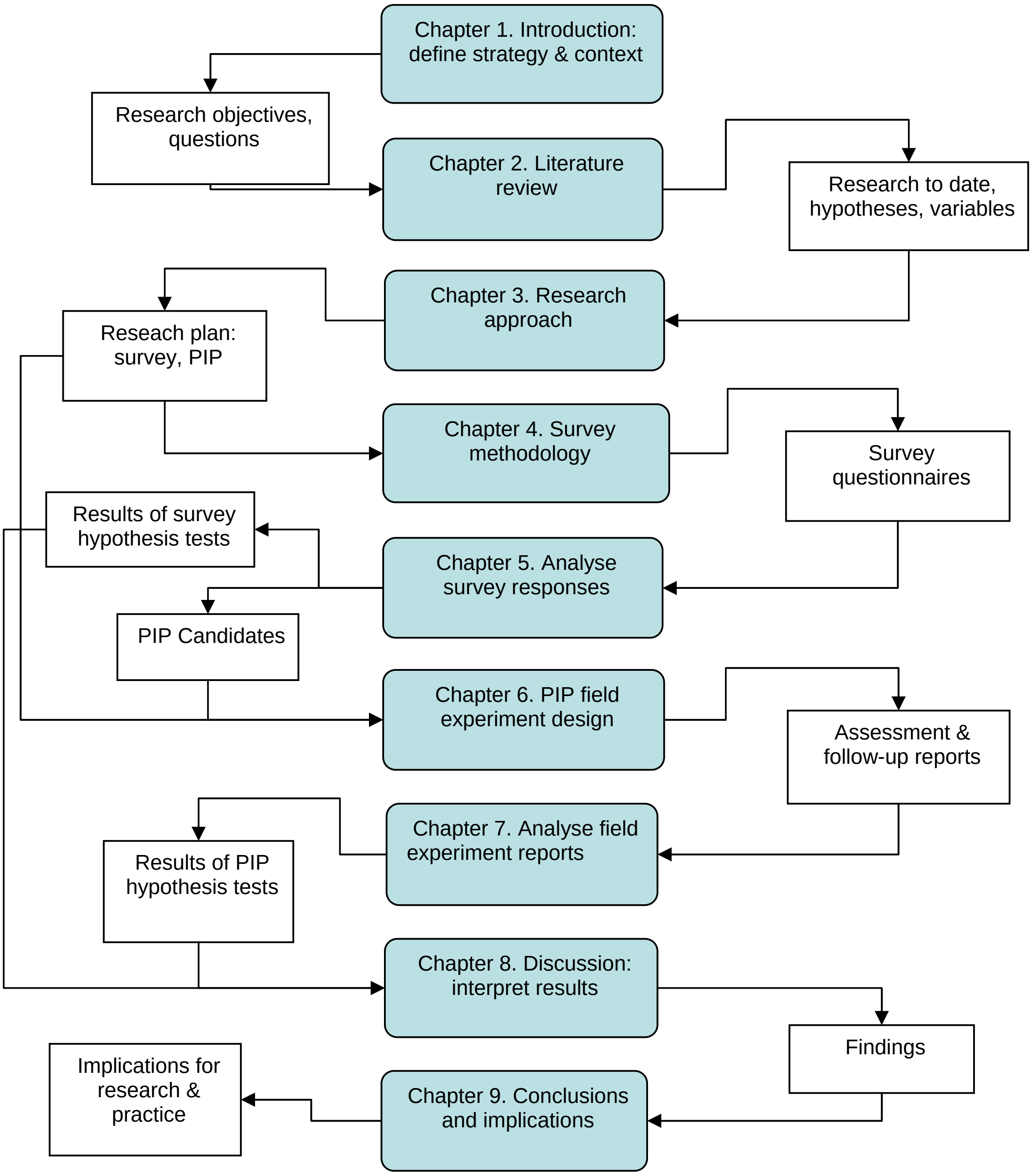


Figure 3-2 Research framework flow chart
(adapted from Gable 1994; Galliers 1991, p. 118)

An overview of the two methods is now provided, the survey method in §3.5.2 and the field experiment method in §3.5.3.

3.5.2 Survey method

The first method used in the study involved the use of a survey as the basis for an exploratory study to clarify the exact nature of the problem and provide data to answer the first research question: is there wide variation in the extent of adoption of software development best practice techniques by software developrs in Queensland, and is adoption related to particular organisational characteristics? Surveys are classified as a type of field study and can provide a cross-sectional picture and quality data about current practices as well as accurately documenting the norm, identifying extreme outcomes and delineating associations between variables (Cornford & Smithson 1996, p. 48; Gable 1994, p. 114). In Chapter 4, the survey method is fully described in terms of its design and execution.

Research can be classified, depending on the primary goal of the study as exploratory, descriptive/predictive and causal (explanatory) (Davis & Cosenza 1988; Sekaran 1992; Yin 1994; Zikmund 1994). This first part of the study, the survey, is classified as exploratory; its main objectives being to determine the extent of adoption of software engineering best practice and to identify appropriate organisations to involve in the field experiments. This stage of the study can further be classified as *correlational*, a form of descriptive study which involves collecting data in order to determine whether, and to what degree, an association exists between two or more quantifiable variables (Gay & Diehl 1992).

A census design using questionnaires and survey techniques is chosen to enable a comprehensive coverage of the software development industry in Queensland in a fast and economical way. These techniques are recommended (by Easterby-Smith, Thorpe & Lowe 1991) as they have 'the ability to describe economically features of large numbers of people or organisations' (p. 35).

However, field experiments, as described in the next section, are also included in the study to overcome known limitations of surveys: the right questions must be asked in the right way; deep understanding is reduced; and discoveries may be inflexible (Gable 1994, p. 114; Kaplan & Duchon 1988, p. 572).

3.5.3 Process improvement program—field experiments

The PIP study, although having some characteristics of a case study, is defined here in this thesis as a field experiment. This term is consistent with the tradition of the European Software Institute in naming their EU-funded interventions Process Improvement Experiments (PIEs). A field experiment is defined by Zikmund as 'an experiment conducted in a natural setting, often for a long period of time' (1994, p. 302). Based on the advice of Pfleeger (1994, p. 17), the current situation in the organisation as determined by the assessment can be considered to be the control, and the new situation as reported at the follow-up meeting is the experimental one. It is recognised that although it is impossible to eliminate disturbing influences in software engineering field experiments, such studies are needed and encouraged, and software projects can be used as large, simple trials studies (Seaman 1999, p. 571).

The variety of experiments reported in journals and conference papers has prompted researchers to develop a range of classification schemes. The field experiments are now described in terms of the research classification schemes developed by Basili (1996), Zelkowitz & Wallace (1996), and Kitchenham (1996).

Considering Basili's (1996) framework in reference to the type of results to be obtained by the study reported in this thesis, this study fits his correlational type as statistical tests are used to assess the change in process capability before and after the software process improvement intervention, and the association between process capability and organisational characteristics. However, the study also plays a valuable descriptive role as the assessment reports describe the current processes of 22 software development firms. Furthermore, the actions taken by each organisation subsequent to the assessment are also described. As this research investigates and describes the impact of the SPI program, in that it describes the characteristics of a phenomenon and the effects of this phenomenon on the software development organisations, it is considered to be descriptive (Zikmund 1994). Therefore, applying Basili's (1996) framework to the study reported in this thesis, it would be classified as a *descriptive/correlational* experiment, involving *expert* subjects (experienced software developers), *in vivo*, with a *multi-project variation*.

According to the classification scheme proposed by Zelkowitz & Wallace (1996), the PIP study would be classed as a replicated experiment. Following the recommendations from the initial process assessment, some of the usual techniques and processes are replaced by the software developers with methods consistent with the RAPID model. The follow-up meetings gather the results to enable the hypotheses to be statistically tested. Applying Kitchenham's (1996) classification scheme to this study, both the best practice survey and the PIP field experiments are of the quantitative evaluation type.

The field experiments carried out for this study involved 22 small software development organisations participating in a process improvement program. The process improvement program reported in this thesis was designed and executed by staff and associates of the Software Quality Institute and funded by SEA Queensland. At the same time, a different study, named the *Showcase* Program was also undertaken by the SQI with two software development companies. The *Showcase* Program included mentoring and more intensive involvement on the part of the SQI assessors/consultants. The *Showcase* Program is not included in this study as this researcher was not personally involved in its design or execution. Furthermore, as the *Showcase* program represented a different type of experiment, the results are not included with the 22 replicated PIP results analysed here.

The author of this thesis participated as an assessor in three of the 22 field experiments and analysed the reports from all the experiments. The process improvement program entailed an assessment of software development processes at each site (pre-test), recommendations for improvements were made, then six months later, a follow-up meeting was arranged at each site to evaluate process improvements (post-test). The follow-up meetings formed an important part of the PIP providing reasons for the differences in the effectiveness of the program (as suggested by Agresti 1993b).

The process improvement program was funded by SEA (Qld) for a selection of its member organisations. Apart from SEA membership and the time commitment (in-kind contribution) of staff at the organisations, the participating organisations incurred no further costs.

3.6 Reference disciplines for methodology

The field of software engineering draws on both natural science and social science for research methodology. As mentioned in §3.2 above, the field of information systems is used as a reference discipline to provide taxonomies to describe the research paradigm, and approach. The concepts of experimentation come from the 'hard sciences' such as physics and chemistry (Zelkowitz & Wallace 1997), and have a strong tradition in engineering research.

3.7 Unit of analysis

A crucial aspect of the problem definition is determining the unit or level of analysis. In their review of software engineering research, Glass, Vessey and Ramesh (2002) recognise that research work is often couched in a social setting and they identify ten different levels (units) of analysis: society, profession, external business context, organisational context, project, group/team, individual, system, computing element (e.g. program), and finally, abstract concept (p. 498). The unit of analysis for this study is *organisation* but focussing on small software development firms, although medium and large organisations are also included in the survey. In comparing this study with other research, the small number of staff employed at some of the firms would be similar to the number of staff involved in a software development *project, group* or *team.*

3.8 Research plan

Now that the research paradigm, approach and methods have been introduced, the actual plan formulated to answer the research questions is described.

The motivation for the study came from the researcher's prior industry experience as programmer, analyst and project manager. This led to the researcher's interest in process improvement in small software development teams. SEA (Qld), through its association with the SQI, sponsored the best practice survey and the process improvement program. These two initiatives provided the researcher with a valuable opportunity to collect and analyse data at both industry and firm level. The SPI literature was reviewed and research questions formulated and refined. The research approach was developed and a two-stage data collection method planned to test the hypotheses:

Stage 1: Best practice survey

Stage 2: Participation in, and analysis of reports from the process improvement program.

The steps involved in the first stage, the best practice survey, are described in Chapter 4, similarly, the steps required for the second stage, the process improvement program are presented in Chapter 6. The timeframe of the study showing activities and dates is included in Appendix C.

The three research questions and accompanying hypotheses are formulated in Chapter 2, and graphically depicted in Figure 2-2, Figure 2-3 and Figure 2-4. Table 3.1 lists the variables involved in each hypothesis test. These variables are defined fully in §4.3 (survey) and §6.9 (PIP), the planned statistical procedures are presented in §4.4 (survey) and §6.10 (PIP), and the actual data and test results in Chapter 5 (survey) and Chapter 7 (PIP).

Table 3.1 Summary list of research questions, hypotheses, and variables

Research Question 1. Is there wide variation in the extent of adoption of software development best practice techniques by software developers in Queensland, and adoption is related to particular organisational characteristics?		
	Independent variable	Dependent variable
H1a		Practice adoption level
H1b	Adoption level of primary lifecycle practices	Adoption level of organisation and support practices
H1c		Organisation adoption level
H1d	Organisation size – number of employees	Organisation adoption level
H1e	Number of software developers	Organisation adoption level
H1f	Development type (COTS etc)	Organisation adoption level
H1g	Industry sector	Organisation adoption level

Research Question 2 Is the capability of technical processes higher than that of management processes, and is process capability is associated with particular organisational characteristics?		
	Independent variable	Dependent variable
H2a	Technical/engineering process capability	Management/organisational process capability
H2b	Organisation size	8 Process capability levels and attribute achievement
H2c	Proportion of experienced staff	8 Process capability levels and attribute achievement
H2d	Proportion of staff with formal educational qualifications	8 Process capability levels and attribute achievement
H2e	Proportion of technical staff	8 Process capability levels and attribute achievement
H2f	Duration of firm operation	8 Process capability levels and attribute achievement
H2g	Target business sector	8 Process capability levels and attribute achievement
H2h	ISO 9001 certification	8 Process capability levels and attribute achievement
H2i	Number of projects	8 Process capability levels and attribute achievement
H2j	Size of project team	8 Process capability levels and attribute achievement
H2k	Project duration	8 Process capability levels and attribute achievement
H2l	Average cost overrun	8 Process capability levels and attribute achievement
H2m	Perceptions of firm performance – 6 measures	8 Process capability levels and attribute achievement
Research Question 3 Are assessment-based SPI programs an effective means to improve process capability for small software development firms, and is the extent of improvement in process capability associated with particular organisational characteristics?		
H3a	Process capability levels and attribute achievement of eight processes before PIP	Process capability levels and attribute achievement of eight processes after PIP
H3b	Firm size – number of FTE employees	Extent of improvement in capability of 8 processes
H3c	Proportion of experienced staff	Extent of improvement in capability of 8 processes
H3d	Proportion of staff with formal educational qualifications	Extent of improvement in capability of 8 processes
H3e	Proportion of technical staff	Extent of improvement in capability of 8 processes

3.9 Conclusion

In this chapter, the research approach was described and the research method was justified. This study used a positivist multi-method research approach. Survey and field experiments were selected as the best approach to explore the research problem and to answer the research questions developed to guide the project. The next chapter focuses on the procedures associated with the design and execution of the survey, and discusses ethical issues related to the survey. The procedures used for the field experiment are detailed in Chapter 7.

4 CHAPTER FOUR—BEST PRACTICE SURVEY METHODOLOGY

4.1 Introduction

In this chapter, a detailed description of the survey procedures is provided. The survey provides data to investigate the first research question. The primary aim of the survey was to determine the extent to which software development organisations are using best practice techniques. The secondary aim was to provide information for the planning stage of the process improvement program. From the survey responses, it was possible to compile a list of candidate software development firms interested in participating in the process improvement program, and also to determine the extent to which best practice was currently in use.

The first section of this chapter builds on the introduction to the methodology provided in Chapter 3, describing the design of the survey and providing evidence that appropriate procedures were followed. In §4.3, the variables sourced from the survey are defined, and §4.4 explains the data analysis used for the statistical tests for the hypotheses related to the survey data. The limitations of the survey methodology are discussed in §4.5, followed by consideration of ethical issues related to the survey (§4.6).

4.2 Survey design

In this section, a detailed description of the survey procedures is provided, including the unit of analysis, questionnaire design, selection of participants, survey test and execution, and recording of responses.

4.2.1 Survey unit of analysis

The unit of analysis for the survey was any Queensland organisation undertaking software development. The target population included all organisations in Queensland which develop software for sale as well as in-house software development groups within organisations. Rather than identify a sample of the population, the aim was to reach the entire population of organisations which develop software. The survey was therefore a *census* as it sought to include all organisations undertaking software development, not just a sample thereof.

4.2.2 Survey design

To overcome constraints of time and cost, and to enable comparison of Queensland developers with those in Europe, the European Software Institute (ESI) was approached for permission to customise and use the ESI software best practice questionnaire (SBPQ). Permission was granted on two conditions: that minimal change was made to the questionnaire; and that the results would be made available to ESI for comparison with the European data. The aim, origin, and previous use of the SBPQ is detailed in §2.4.

Practice adoption level and organisation adoption level are the dependent variables and care was taken to maintain consistency in terms of scales and items to ensure that comparisons with previous studies could be made. A mail survey was chosen as the data collection method to provide a quick, inexpensive, efficient, and accurate means of assessing information about a population (Zikmund 1994).

4.2.3 Survey pre-test

Prior to the data collection, the survey instrument was pre-tested to enable clarification of constructs; to provide the means of operationalising selected constructs; and because pre-tests can be useful in 'qualitatively establishing the reliability, construct validity, and content validity of measure' (Straub 1989, p. 162). In order to locate and correct weaknesses in the questionnaire, the questionnaire was pre-tested using face-to-face interviews with two key informants, and further feedback was received from an SEA Board member. The selection of interviewees for these pre-tests was designed to obtain maximum feedback from software developers in various roles. Participants included experienced systems analysts from specialist developer organisations, and the then Chief Executive Officer (CEO) of SEA (Qld).

Prior to the pre-test, the following checklist (adapted from Dillman 1978; Revenaugh 1992) was used to review the questionnaire instrument:

- Will the words be uniformly understood?
- Do the questions contain abbreviations or unconventional phrases?
- Are the questions too vague?
- Is the question too precise, biased or objectionable?
- Is it a double-barrel question?

- Does it have a double negative?
- Are the answer choices mutually exclusive?
- Has too much knowledge been assumed?
- Is the question technically accurate?
- Are the questions too cryptic?

During the pre-test, concerns were raised about the section headings and question groupings of the ESI questionnaire (such as *metrics*, *standards and procedures*, *control of development process*). Heeding Dillman's (1978) warning that more technically complex questions may evoke feelings of anxiety or inadequacy, it was decided to group the questions in more of a software development lifecycle sequence so that less experienced software developers would not be intimidated by the section headings, and thus respond more readily. The five new headings used were *Requirements and Design*; *Code and Test*; *Configuration Management*; *Estimates and Schedules*; and *Project Management and Training*. Appendix D Table D.1 provides a mapping of the best practice survey questions to the original ESI questionnaire.

During the pre-test, the respondents completed the questionnaire in the presence of the researcher, and identified any difficulties with interpretation of words or questions. As well as testing the reliability and construct validity, the pre-tests served as 'dry runs' for the final administration of the instrument (Straub 1989). Following Fowler's (1988) advice, emphasis was placed on identifying questions that respondents may misunderstand, misinterpret or find difficult to answer, as well as format or design problems. Particular attention was focussed on the instructions, both in the cover letter and also those embedded in the best practice survey.

Two additional questions were included in the body of the questionnaire to provide information relating to the use of programming languages and development tools. SEA (Qld) requested this information to help tailor their forthcoming training program to industry requirements. The information about the use of programming languages and development tools contributes to building a profile of software developers in Queensland. The list of programming languages was sourced from the International Software Benchmarking Standards Group (ISBSG 1998).

To further customise the instrument to local conventions, the Australian and New Zealand Standard Industrial Classification (ANZSIC) list of industrial sectors was used in place of the European sector breakdown list. Appendix D Table D.2 provides a mapping the ANZSIC codes to the industry codes used in the ESI questionnaire.

The format of the questionnaire was changed to appear more compact, and the few double-barrel questions were split to reduce ambiguity. Finally, the format was changed so that the questionnaire fitted onto two sides of one A3-sized page, with a centre fold creating a four page A4-sized booklet. This was done to appeal to respondents as the four page booklet style looked like it could be completed quickly. Research indicates shorter questionnaires are associated with higher response rates (Bogen 1996). Included in the survey were instructions for respondents, presented in italic font. These comments were used to give direction for using the scale, or were specific to the particular construct, providing a common frame of reference (Spector 1992, pp. 26-7).

4.2.4 Questionnaire format

The first part of the questionnaire sought demographic information. The second part listed the 44 questions pertaining to the use of software development practices, and two questions about programming languages and tools. The third and final part provided space for respondents to record which aspects software development needed improvement and to suggest how SEA could help. A summary of the composition of the questionnaire is shown in Table 4.1. A copy of the questionnaire is included in Appendix E. The variables sourced from the survey are defined §4.3.

Table 4.1 Composition of best practice questionnaire

Part	Section	No. of questions	Purpose of questions
A		7	Demographic and organisational characteristics
B	1	6	Software practices – requirements & design
	2	15	Software practices – code & test
	3	5	Software practices – configuration management
	4	9	Software practices – estimates & schedules
	5	11	Software practices – project management
C		2	Aspects of software development needing improvement, how can SEA help?

4.2.5 Selection of survey participants

As previously mentioned in §4.2.1, the unit of analysis was Queensland organisations undertaking software development. The target population was all organisations in Queensland which develop software for sale (custom and COTS developers) and in-house software development departments in large organisations. A single list of all organisations undertaking software development was not available, therefore it was impossible to undertake formal sampling of the population. An attempt was made to reach the entire population of organisations which develop software, in effect, a *census* of all Queensland software developers.

A list of 5,600 likely organisations was sourced from databases listing organisations and contact lists from the Queensland Government's Information Industries Bureau (IIB), SEA (Qld) and the SQI. To select the in-house developers, the MIS 3,000 database was used (Gartner Group 1995). All Queensland organisations with at least 100 computer screens and five IT staff were selected, providing a contact list of 168 organisations. The *Australia on Disk* database (Dependable Database Data 1998) contains the listings of the Telstra Yellow Pages telephone directory. Each business record includes the Australian and New Zealand Standard Industrial Classification (ANZSIC) code. Codes relating to software development firms are shown in Table 4.2. The codes selected for this study form part of the subdivision 78: Business Services, within Division L: Property and Business Services.

Table 4.2 Australian & New Zealand standard industrial classification

ANZSIC Code	Description
783	Computer services
	Data processing services
	Information storage and retrieval services
	Computer maintenance services
	Computer consultancy services

(Source: ABS 1993)

From the Australia on Disc database, all Queensland organisations with an ANZSIC code commencing with 783 were selected. Any organisation without a valid postal address was removed from the list of selected records. This yielded 2,631 name and address records. The IIB provided a list of approximately 10,000 organisations, and 200 contacts were provided by the SQI. In order to merge these four lists into one mailing list, all the lists were imported into the GoldMine software package, and de-duplicated on the basis of telephone number. As the format of the phone numbers varied depending on the source of contact, the list was printed in alpha name sequence and manually checked by the researcher to eliminate further duplicates. The final mailing list comprised 5,634 records.

4.2.6 Survey execution

The survey, which formed part of the Australian National Industry Improvement Program (NIIP), was supported by SEA Queensland and the Software Quality Institute (SQI) of Griffith University. The intention was to conduct the survey initially in Queensland, then later in other Australian states.

A cover letter for the survey was prepared and printed using the SEA (Qld) letterhead stationery. Where a contact name was recorded, the cover letters were personally signed by the SEA (Qld) CEO, letters with only organisation details were addressed to *The Software Manager* and a scanned signature printed. The text of the cover letter and copy of a questionnaire are included in Appendix E. In January 1999, 5,634 questionnaires were mailed with a cover letter and reply-paid envelope to organisations identified as possible software developers. Of the sent questionnaires, 525 were returned as ‘undeliverable’, all these addresses were checked using the Telstra White Pages web site and 117 of these questionnaires were re-addressed and sent to the updated addresses.

In order to boost the number of responses, a targeted follow-up was conducted. During March 1999, SEA (Qld) engaged a telemarketer to contact 450 organisations from the IIB list. A further 200 questionnaires were remailed on request as a result of this activity.

4.2.7 Recording survey responses

A web-based system was developed to facilitate data entry of responses. As well as facilitating multi-user data entry, this interface enabled provision of feedback to respondents. In the future, the survey can be conducted as a web-based survey in other States, thereby reducing the time, printing and postage cost incurred with paper-based questionnaires, letters and envelopes. For this Queensland survey, respondents filled out a hardcopy form and returned by reply-paid post. The web version was used as a data entry interface to facilitate the capture of the survey responses.

Data files of responses were then generated from the web-based system and imported into MS Excel and SPSS for analysis. The results of the analysis of the data collected are detailed in Chapter 5.

After the survey had been completed, all the survey respondents were sent an email advising them of a URL, password and procedure to access a web-site containing their response and a summary of all the responses. This feedback enabled respondents to benchmark their development practices against the aggregated responses of all the survey responses.

4.3 Definition of survey variables

The survey provided the following variables to be used in the hypothesis testing for the first research question: practice adoption level; organisation adoption level; organisation size; development group size; type of development; industry sector.

Practice adoption level. For each of the 44 questions about best practice, possible responses were yes, no, not applicable or blank. These were keyed into the web-based system as Y, N, U or left blank (null). In this part of the questionnaire, 35 questions offered a two point *yes/no* scale, the remaining nine questions offered three options: *yes; no; or not applicable*. The nine items with the *not applicable* option include an 'applies to' clause to provide explanation of cases when the technique or practice would apply.

To calculate the *practice adoption level*, for each of the 44 practices the number of *yes* responses from respondents were counted and a percentage calculated based on the proportion of *yes* responses to the sum total of *yes* and *no* responses. For the nine questions with the option of *not applicable*, responses indicating the practice was *not applicable* were omitted from the calculation and treated the same as missing values. Practice adoption level is treated as an interval type variable with a range from zero (none of the respondents use the practice) to 100 percent (all respondents use the practice). It is accepted practice to consider adoption level as measured on an interval scale. Reports based on the ESI questionnaire use parametric analysis methods (Dutta, Kulandaiswamy & Van Wassenhove 1996; Dutta, Lee & Van Wassenhove 1998; Dutta & Van Wassenhove 1997a; Dutta, Van Wassenhove & Kulandaiswamy 1998; ESI 1995, 1996b, 1996a, 1997).

Organisation adoption level. For each organisation the number of *yes* responses to the 44 best practice questions was counted and a percentage calculated based on the proportion of *yes* responses to the sum total of *yes* and *no* responses. This is consistent with the calculation used by the ESI:

> 'in the study, *adoption level* represents the percentage of 'Yes' responses relative to the number of 'Yes' or 'No' responses for an individual or group of questions. For a given company, the overall adoption level is computed by looking across all the questions in the questionnaire. To compute the adoption levels across sectors or countries, the individual company adoption levels are averaged' (Dutta, Lee & Van Wassenhove 1998, p. 3).

Organisation adoption level is an interval type variable representing the proportion of adoption with a range from zero (none of the 44 practices used) to 100 percent of practices in use.

Organisation size and development group size are recorded by the respondent in response to the item *number of employees* and *number of software development employees*. This study uses the Australian Bureau of Statistics (2002) definition of organisation size based on the number employees. A small organisation employs less than 20 staff, organisations with 20-199 employees are classed as medium sized. For this study, two further categories are defined: large business with between 200 and 5,000 employees, and more than 5,000 are designated as very large business.

For software development group size, a different categorisation is used. Small development groups comprise less than five staff; medium groups have between five and nine developers, and large groups have more than ten staff.

Type of development. Seven variables were defined to record involvement in the software industry (item A3 of the questionnaire): software user—developed in-house;

software user—developed by a 3rd party; software developer—producing off-the-shelf systems; software developer—producing custom software systems; research and development institute or university; interest group—professional society or standards body; other. For each response, a tick in the box indicated a yes response and blank indicated no. Each respondent could indicate that their organisation was active in more than one type of involvement. From this item on the questionnaire, two main variables of interest were derived: *COTS developer* representing software developers producing off-the-shelf systems; and *custom developer* representing software developers producing custom software systems. These variables are measured on a nominal scale, with a value of 1 for yes, 0 for no.

Industry sector. As mentioned in §4.2.3, in formulating the survey, the ESI list of industry sectors was replaced with the ANZSIC list. Respondents could choose from a list of 20 industry sectors, or chose the option of other and record a comment. Industry sector is a nominal type variable with a range of 20 values.

4.4 Data analysis method for survey responses

In this section, software used for the statistical tests is described, and the statistical tests used for each hypothesis are named and justified. To enhance readability, a glossary defining specific statistical techniques used in this thesis is included in Appendix B.

Data gathered from the survey were imported into MS Excel, then SPSS. Prior to undertaking any correlation analysis to statistically prove the association between variables such as organisation size and adoption of best practice, the characteristics of the data were explored in order to ensure the correct statistical approach was selected. As the assumption of normality is a prerequisite for many inferential statistical techniques, the distributions of both adoption level variables were checked for normal distributions. Descriptive statistics such as calculations of means and standard deviations, and cross tabulations were performed. The Pearson product moment correlation coefficient was calculated to describe the relationship between interval scaled variables (practice and organisation adoption levels with organisation size and size of the software development team). Students t tests were used to compare the means of adoption levels of groups of responses in relation to practice type and industry involvement type. To compare the adoption level across different industry sectors, ANOVA tests were used.

To identify statistically significant differences or associations the confidence level is set at .05 and all statistical tests are reported accompanied by the corresponding probability level, for example, p=0.03.

4.5 Limitations of the survey methodology

In this section, limitations relating to the survey data collection methods and data analysis procedures are discussed. The limitations related to the PIP field experiment method are discussed in §5.12. Validity issues relevant to the survey include construct, statistical and external validity. One of the acknowledged difficulties with the use of self-administered surveys, such as the best practice survey, is the lack of control by the researcher of the environment in which the respondent completes the survey. An attempt to address this concern was made by encouraging potential respondents to complete the questionnaire conscientiously, firstly by keeping the survey instrument as short as possible and secondly, by providing feedback in the form of a web-based summary of the survey results to respondents.

4.5.1 Construct validity

Construct validity concentrates on 'the fit between constructs and the way that the research problem is conceptualised' (Cook & Campbell 1979, p. 64). As detailed in §4.2 and §4.3, care was taken to select, review and tailor an existing instrument validated by other researchers who had investigated a similar research problem. Also, as described in §4.2.3 comprehensive pre-testing of the survey was undertaken as recommended by Straub (1979). Limitations are also recognised regarding the ability of the best practice questionnaire to adequately measure the adoption level of best practice techniques. As explained in §4.2.2, the questionnaire was based on the instrument developed by the ESI. The ESI administered the questionnaire to organisations applying for funding for process improvement programs; it was not designed to be used as an industry survey or census. Consequently, the accuracy of the variable *adoption level* is limited by the choice of practices included in the questionnaire. It is recognised that the list of practices may be incomplete, may include outdated practices, or may be biased towards the practices used by a particular segment of the software industry.

4.5.2 Statistical conclusion validity

Care was taken to ensure that false conclusions were not made from the data analysis. Firstly, rather than exploring in the data for relationships, this study concentrated on testing and reporting hypotheses which had been derived from theory. Also, when the statistical tests are reported, the assumptions underlying each of them are discussed and care was taken that the underlying assumptions were not violated (Kitchenham et al. 2002). Furthermore, as advised by Perry, Porter and Votto (2000), the confidence level of each statistical test is reported.

4.5.3 External validity

External validity refers to the ability to generalise the results of the research to the external environment (Zikmund 1994, p. 306), and can be improved through replication of the study in other places, with different people, at another time. This study replicates the ESI survey which was conducted on three occasions (1995-1997), and included organisations from many different European countries. Therefore, the ESI results can be generalised to European software development organisations requesting EU funding.

However, for this study, the extent to which the findings can be generalised to Queensland software development organisations hinges on whether the respondents are representative of software development organisations in Queensland. By using a census rather than a random sample, sampling frame errors were avoided. The extent of representativeness was determined by comparing characteristics of the responses against the Queensland software industry profile (§5.3.7). The responses included in-house as well as commercial developers; the proportion of small organisations was consistent with the industry proportion of small firms; and regional and remote organisations were represented in the responses. Therefore, the responses appear to present a balanced coverage of a wide range of software development organisations, with a possible bias towards those interested enough to take the time to complete the questionnaire. As the Queensland software industry is similar to the wider Australian industry, it could be argued that the results from this study may be generalised to that context as well.

4.5.4 Reliability

In regards to the survey, respondents may have knowingly given untrue or misleading answers (Emory, 1980 cited in Moore 1989, p. 236). Khurana et al. (1996) assert that it is difficult to obtain reliable data about current practice because developers are reluctant to admit that they are producing poor software. It is also possible that some survey respondents experienced *evaluation apprehension*. Fearing failure, respondents may have wished to show their organisation in a more positive light, or may have provided the ‘correct’ answers, in other words, the responses they thought the researcher was seeking.

Although the best practice questionnaire was addressed to the Software Manager by name or by title, the researcher had no control over who actually completed the form. Also, it is possible that some respondents may not be aware of the actual extent of use of various software engineering practices within their organisations. Weinberg (1995) and Wilson, Petocz and Roiter (1995) found that staff at different levels in software organisations reported inconsistent results when surveyed about actual use of software development practices.

In summary, while a conscientious effort was made to ensure that the data collected were as accurate and reliable as possible, a number of limitations have been identified. Constraints imposed by time and cost precluded a full resolution of these limitations.

4.6 Ethical considerations related to survey

The ethical issues raised by Zikmund (1994, p. 257) are now considered in regards to the survey. The survey did not breach the respondent’s right to privacy as the organisation contact details were publicly available. The purpose of the research was clearly and frankly stated in the cover letter, and there was no deception used. Furthermore, the respondents’ rights in regard to confidentiality were recognised and assured. Care was taken to check and accurately process all survey responses, and the data were reported objectively and checked by the researcher’s supervisors.

Although confidentiality was considered to be an important issue, respondents were requested to provide their name, organisation and contact details. There were two reasons for dispensing with anonymity: to enable a follow-up of non-responses; and to send an email with the URL and password to respondents to inform them how to access the summarised feedback. Confidentiality was assured in the covering letter, as it was anticipated that some organisations may not appreciate details of their software engineering practices being made public.

It was assumed that honest responses were provided from survey respondents and participants. Also, participating organisations were provided with the opportunity to compare their responses with the overall findings. Indeed, the incentive of access to a summary of the results of the survey may have encouraged participation in this research (Kalafatis & Tsogas 1994).

4.7 Summary

In this chapter, the survey instrument and procedures were described. An existing *software best practice* survey instrument was adapted and pre-tested to ensure maximum validity and reliability. A description of the sampling plan was provided. The survey was aimed at all Queensland-based organisations that develop software. After the responses were checked, they were keyed into a web-based system and exported to MS Excel and SPSS for statistical analysis. The variables sourced from the survey were defined, and limitations in terms of validity and reliability considered. Ethical issues related to the survey were also discussed.

In the next chapter, the survey response is described with a profile of the respondents before presenting a detailed account of the statistical analysis undertaken to test the hypotheses related to research question 1.

5 CHAPTER FIVE—SURVEY RESULTS

5.1 Introduction

In the previous chapter, a description was provided of the survey which was conducted to collect data about the adoption of best practice techniques by software development organisations in Queensland. This chapter uses three main sections to report on the analysis of that data. Firstly, the response to the survey is reported, and then the profile of respondents is presented. The analysis of the survey data is reported next, with particular emphasis on exploring the distribution of the variables representing the extent of adoption of best practice, and their association with organisational characteristics. Finally, a summary of the results of the hypothesis tests relating to the survey is provided. The interpretation and implications of the findings presented in this chapter are discussed fully in Chapter 8.

5.2 Survey response

When the completed questionnaires were received, they were checked and then keyed into the web-based system described in §4.2.7. Survey data were imported into MS Excel from the web system and, after preliminary checking for completeness, transferred into the statistical package SPSS. This software was used to provide descriptive statistics, assess normality, and perform statistical analysis.

As detailed in §4.2.6, questionnaires were posted in January 1999 to 5,634 Queensland organisations selected from multiple sample sources. From this mailout, 209 responses were received from software developers, 354 responses from organisations indicating that they did not develop software, and a further 408 undeliverable envelopes returned (refer Table 5.1 for detailed breakdown of returns).

Table 5.1 Breakdown of survey distribution and return

Survey Distribution and Return	Number of Questionnaires
Questionnaires posted 1st mail-out Jan 1999	5,634
Returned as undeliverable	525
Questionnaires remailed to corrected addresses Mar 2000	117
Valid responses from software developers	203
Invalid responses from software developers	6
Responses from organisations not involved in software development	354
Undeliverable – correct address not found	408
Total returned	974

The respondents who answered *No* to the question *Does your organisation develop or maintain software?* were excluded from further analysis as they did not represent the unit of analysis, viz. firms undertaking software development. There were four respondents who said they developed software but did not complete the questionnaire: these were deemed invalid responses. In addition, two people responded twice: it was assumed they wished to correct information on their initial response, therefore their first response was deleted and deemed invalid, and the later response was included. To determine the effective response rate, de Vaus (1996) advises basing the calculation on the proportion of eligible organisations who received the questionnaire. From the 5,634 organisations to whom questionnaires were mailed, 408 were returned to the researcher by Australia Post as undeliverable, therefore 5,226 were received by organisations. Of the 563 questionnaires completed and returned, 62.9 percent (354) were out of scope (i.e. non-developers). Applying this proportion to the number of organisations who received questionnaires, it is estimated that 3,544 would have been non-developers, leaving 2,090 in-scope sample elements. Of these, 209 responded, giving an effective response rate of 10 percent.

The response rate is disappointing but within the typical range of 10-20 percent for mail surveys to business establishments (Paxson 1992). Businesses have built up a considerable resistance to answering mail surveys due to constant inundation with both mail and telephone surveys (Attewell & Rule 1991, p. 306). Recent large scale surveys of software development practices by Stalhane, Borgersen and Arnesen (1997) and Larsen and Kautz (1997) received response rates of 8.4 percent and 13.3 percent respectively.

5.3 Respondent profile

In this section, a descriptive analysis is provided of the survey responses in terms of geographical distribution, business sector, organisation size, development team size, and primary involvement in software development industry.

5.3.1 Geographical distribution

In order to enable analysis of the geographical range of the respondents, the region names were determined consulting the Print Post Sort Plan (Australia Post 1999) using the postcodes provided by respondents. Analysis of the responses based on the respondent's postcode, as shown in Table 5.2, revealed that most of the organisations were located in, or in areas adjacent to, the state capital city. However, 15 percent of the responses came from regional areas.

Table 5.2 Geographical distribution of responses

Region	Frequency	Percent
Brisbane	138	67.98
Cairns	3	1.48
Gold Coast	24	11.82
Ipswich	2	0.99
Mackay	1	0.49
Rockhampton	3	1.48
Sunshine Coast	9	4.43
Toowoomba	16	7.88
Townsville	7	3.45
Total	203	100.00

5.3.2 Industry sector

Prior to analysing responses relating to industry sector, respondents who chose the 'other' category and recorded their industry type were examined. The relevant industry sector was determined from the organisation's web site and these were then recoded into the appropriate industry sector. The full list of industry sectors and frequency of responses is included in Appendix F Table F.1. Most industry sectors were represented. Frequencies and percentages were calculated for the responses, highlighting that responses from the Software Development sector formed the largest group, accounting for more than three quarters of all responses, followed by Information Technology (10 responses), Education (7); Manufacturing (5); Utilities (5); Government Administration and Defence (5); Health and Community Services (3); and two responses each from the following sectors: Agriculture, Forestry and Fishing; Mining; Communication Services and Media; and Finance and Insurance. Only one response was received from each of these sectors: Construction; Retail and Wholesale; Property and Business Services; and Cultural and Recreational Services. No responses were received from the following sectors: Accommodation, Cafes and Restaurants; Transport and Storage; Personal and Other Services; and Tourism and Hospitality.

5.3.3 Organisation size

Two questions relating to organisation size were included in the survey. The first question asked the total number of employees in the organisation and the second the number of employees involved in software development or maintenance.

As detailed in §4.4, the Australian Bureau of Statistics categorise organisation by size on the basis of number of staff. Table 5.3 shows the breakdown of number of responses by organisation size category: almost half of the Queensland respondents were from micro-businesses with less than five total staff, and almost three quarters of respondents have less than 20 employees. However, larger organisations were also well represented with 34 respondents in the medium category with 20 to 200 staff, 22 organisations reported in the large category (200-5,000), and three very large organisations with more than 5,000 employees.

Table 5.3 Summary of responses by organisation size

Organisation Size	No of employees	Frequency	Percent	Cumulative percent
Micro-business	<5	91	44.83	44.83
Small	5-19	53	26.11	70.94
Medium	20-199	34	16.75	87.69
Large	200-4,999	22	10.84	98.52
Very Large	5,000+	3	1.48	100.00
Total		203	100.00	

5.3.4 Software development group size

As well as the total number of staff in each organisation, respondents provided the number of staff involved in software development. Using the categories outlined in §4.3, Table 5.4 shows that most of the responses (68%) were from organisations employing less than five software developers. Only 15 percent of firms reported ten or more software developers.

Table 5.4 Summary by number of software developers

Development group size	Number of software development employees	Frequency	Percent	Cumulative percent
Small	0 staff	2	1.0%	1.0%
	<5 staff	136	67.0%	68.0%
Medium	5-9 staff	34	16.7%	84.7%
Large	10-25 staff	12	5.9%	90.6%
	25-50 staff	11	5.4%	96.0%
	50-100 staff	7	3.4%	99.4%
	100-250 staff	1	0.5%	100%
	>250 staff	0	0%	
TOTAL		203	100.0%	

5.3.5 Primary involvement in software industry

It was recognised that organisations may play a variety of roles in the software industry. For example, most large organisations would develop and use in-house systems as well as purchasing 3rd party software. Also, many software development companies produce off-the-shelf packages as well as providing custom developed systems to clients. The survey included a question to determine the primary involvement of organisations in the software industry. Although the question was worded to encourage respondents to choose only one option as their primary involvement, the results (in Table 5.5) show many respondents selected more than one option.

Table 5.5 Distribution of responses by primary involvement in software industry

Organisation's primary involvement in industry	Frequency	Percent
Software user – developed in-house	53	26.1%
Software user – developed by a 3rd party	43	21.2%
Software developer producing off-the-shelf systems (COTS)	87	42.9%
Software developer producing custom software systems	128	63.1%
Research and development institute or university	10	4.9%
Interest group e.g. professional society or standards body	2	1.0%
Other	7	3.4%
Total options selected Average options selected per respondent	328 1.6	

As highlighted in Table 5.6, there was considerable overlap in the User and Developer categories, in fact only 61 organisations were not primarily users and 37 were not primarily developers. This question should have enabled identification of the software development firms from other organisations who develop or use software in their business. However, it was not possible to separate responses in this manner as 30 respondents chose both categories: software user as well as software developer. The two categories for software developers are abbreviated throughout this thesis: COTS (commercial off-the-shelf) refers to respondents who selected their primary involvement as a software developer producing off-the-shelf systems; custom refers to those organisations producing custom software systems.

Table 5.6 Analysis of frequency of responses by 'Software User' and 'Software Developer'

Developer type	Count yes/no	Software user developed in-house		Software user developed by a 3rd party	
		Yes 53	No 150	Yes 43	No 160
Developer—COTS	Yes 87	10	77	10	77
	No 116	43	73	33	83
Developer—custom	Yes 128	23	105	18	110
	No 75	30	45	25	50

There were six firms who did not choose any of the four user/developer options. Two organisations selected research and development, two chose other (accounting software and testing software) and two respondents did not tick any of the boxes offered. In total, seven respondents selected the option of 'other', recording their primary involvement as: software for accounting practices; web development; computer systems consultants; testing software; computer retail; and multimedia production. Five of the respondents who selected 'other' also selected another option.

5.3.6 Programming languages and tools

The questionnaire provided a list of 14 programming languages to determine which languages are in current use and planned for future adoption. Only four organisations left this question blank. Overall, on average, three programming languages were recorded for each response. The list of languages and frequency of responses is provided in Appendix F, Table F.2. The most popular languages were Visual Basic (43%), MS-Access (36%), C++ (31%) and SQL (31%). One third of respondents reported a language other than the 14 listed, in fact, 41 additional *languages* were recorded, indicating an extremely diverse range of programming languages in use by this group of organisations. Table F.3 in Appendix F lists the names and frequencies of the 'other' programming languages. As well as recording current programming language used, organisations were asked which languages they planned to use in the next 12 months. The use of Java was expected to rise from 17 percent to 26 percent, with the use of all other languages expected to decrease.

As far as development tools, the questionnaire listed seven types of tools. Ten percent of respondents reported tools other than the seven types offered. A detailed summary of responses to this question, and a full list of the other responses are included in Appendix F, Table F.4 and F.5 respectively. The tools listed in the questionnaire were types of tools, for example defect tracking, but respondents often provided specific applications, such as MS Access Issues Register. More than one quarter of organisations did not respond to this question (27%), those that did respond, selected, on average, two tools each. The most popular responses were staff time-sheet system (34%), source code control (25%) and 3rd party components (24%). It was surprising to find that in the next 12 months, many organisations indicated they would cease to use staff time-sheet systems (decrease of 15%), and 3rd party components (decrease of 7%).

5.3.7 Representativeness of responses to industry

As mentioned in §4.5.3, in order to generalise the findings of the survey to the population of software development organisations in Queensland, the demographics of the sample were compared with the industry statistics. As detailed in §4.2.5, to ensure in-house software development groups were included in the survey, 168 organisations were selected from the MIS3000 database as fitting the criteria of at least 100 computer screens and five IT staff. As shown in Table 5.7, 21 of these 168 organisations responded to the survey. A further 10 responses were identified as being received from non-software development organisations (for example, government departments). Therefore, 172 of the responses came from commercial software development firms.

Table 5.7 Responses by source for survey mailing list

Type of Organisation	Number of Responses
Non-software development organisations from MIS list	21
Other non-software development firms (from IIB, SQI lists)	10
Commercial software development/IT firms	172
Total	203

Analysis of the number of employees of the 172 commercial software development firms revealed a mean total number of employees of 15 and standard deviation of three employees. The proportion of these firms with less than 20 staff is 83 percent. This is within the range of 70 percent to 96 percent of Australian software development firms with less than 20 staff estimated by SEA National (2004) and Houghton (2003) respectively.

Although statistics of the geographic spread of software development organisations in Queensland is not known, further assurance of generalisability can be taken from the fact that 15 percent of the respondents to the survey were from various locations other than Brisbane (as detailed in §5.3.1). Therefore, the responses represent organisations across the entire State of Queensland, not solely the capital city.

5.4 Extent of adoption of best practice techniques

In this section, data analysis relating to the first research question is presented: is there wide variation in the extent of adoption of software development best practice techniques by software developers in Queensland, and is adoption related to particular organisational characteristics? Previously, in §4.2.4, the questionnaire was described. The questionnaire comprised 44 questions asking if a particular practice or technique was used in the organisation. Respondents had the option, for each of the 44 questions, to select yes, no, not applicable or not answer the question. A list of the questions, and the frequency of responses for each question (yes, no, not applicable, missing) is included in Appendix F, Table F.6. Overall, 43.76 percent of questions were answered *yes* to indicate the best practice was used; 47.86 percent *no*; 3.78 percent selected as *not applicable*; and 8.37 percent of questions were not answered. The low rate of *not applicable* and missing responses indicates the respondents understood the context of the questions and conscientiously completed the questionnaire. To determine the overall adoption level, the proportion of *yes* responses to the sum of the count of *yes* and count of *no* responses was calculated. As shown in Appendix F, Table F.6, the overall adoption level was calculated as 47.76 percent, indicating that on average, less than half of the practices are used.

In the next section, the adoption level of each practice is examined, then in §5.4.2, the adoption level of each organisation is analysed, and finally the association between the organisation adoption level and organisation characteristics is explored (§5.4.3-§5.4.6).

5.4.1 Analysis of practice adoption level

Prior to undertaking any statistical analysis of the level of adoption of the practices, the characteristics of the data were explored in order to ensure the correct statistical approach was applied. For each of the 44 practices, the practice adoption level was calculated. A full listing of the 44 practice questions, mean adoption level of each practice and number of responses for each question is included in Appendix F, Table F.6. Based on the adoption level of each practice, the mean adoption level of the 44 practices is 47.80 percent and the standard deviation is 22.31 percent (details of distribution in Table F.7 in Appendix F). The assumption of normality was confirmed for *practice adoption level* (Shapiro-Wilks statistic=.955, *p*=.083, test results included in Appendix F Table F.8). The distribution of the variable *practice adoption* is represented graphically by a histogram in Figure 5-1.

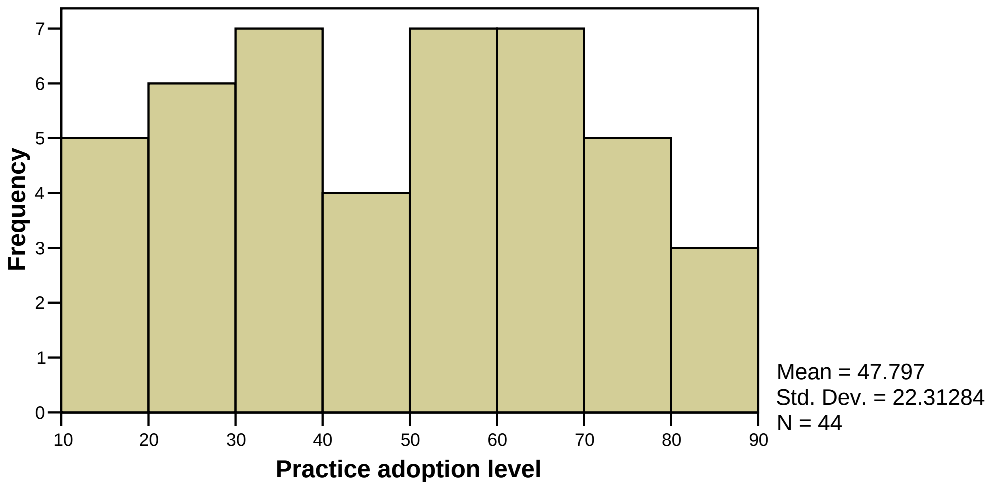


Figure 5-1 Histogram showing distribution of practice adoption level

As evidenced by the standard deviation of 22 percent adoption, the variation in the range of adoption of the 44 practices was wide. The range of adoption of practices extended from the most commonly used technique: *software projects have nominated project managers* for 87 percent of respondents (n=201), to the least adopted practice:

gathering statistics on test efficiency, practiced by only 11 percent of respondents (n=198).

A detailed comparison of the Queensland survey results compared to those from the ESI surveys is provided in §8.2.1. Briefly, the mean of 47.80 percent is only slightly less than that reported from the ESI 1995 survey—48.8 percent; and the standard deviation of 22 percent indicates a greater variation than that from the ESI study—19.6 percent (derived from Dutta, Kulandaiswamy & Van Wassenhove 1996). The reports based on the ESI surveys stated that wide variation of practice adoption existed. As the standard variation of the Queensland responses is in excess of that reported from the ESI survey, adoption of best practice by Queensland developers can also be classified as wide. *Therefore, the null hypothesis H1a, that there is no variation in the adoption of best practices, is rejected.*

A list of the 44 practices in decreasing order of their adoption level is included in Appendix F Table F.9. From this list, the 44 practices were summarised by the extent of their adoption as shown in Table 5.8. Seven of the practices are widely adopted by more than 75 percent of respondents, and at the other extreme, ten practices are only in use at less than one quarter of organisations. The adoption of specific practices is discussed in detail in Chapter 8.

Table 5.8 Summary of extent of adoption of techniques

Extent of adoption range	Number of practices in adoption range	% of practices
75-100%	7	16%
50-74.9%	15	34%
25-49.9%	12	27%
0-24.9%	10	23%
Total	44	100%

Grouping practices by lifecycle phase. As shown in Appendix D Table D.3, the 44 practices of the best practice questionnaire were mapped to the process groups of the ISO 15504 standard, which is based on the ISO 12207 software lifecycle processes standard (ISO/IEC 2002). The mean adoption level for each group of practices was then calculated as shown in Table 5.9.

Table 5.9 Mean adoption level of practices grouped by ISO 15504 process groups

ISO 12207 process group	ISO 15504 process group		Number of practices	Mean adoption level	Std dev adoption level
Primary lifecycle processes	CUS	Customer	4	69.9%	13.2%
	ENG	Engineering	9	50.8%	27.2%
Organisational processes	MAN	Management	13	49.3%	20.5%
	ORG	Organisation	2	21.2%	0.3%
Support processes	SUP	Support	16	42.7%	20.0%
		Overall	44		

According to ISO 12207, the primary lifecycle processes include the engineering and customer process groups, and the other processes are classified as organisational and supporting processes (as shown in Table 5.9). To explore the adoption of practices related to the primary lifecycle processes as compared to the support and organisational processes, the adoption level of the combined customer and engineering process group practices was compared to the adoption level of the combined management, organisational and support process group practices.

Prior to using a t-test to determine if differences exist between the two groups of practices, the assumptions for this test were checked. Firstly, groups must be independent; secondly, the groups should come from populations with equal variances. The first assumption was met as the practices belonged either to the primary lifecycle group of process or the other group, not both. To check the second assumption, Levene's test was used to check for equality of variance. This test confirmed that the population variances are relatively equal (Levene statistic based on mean=.739, df=42, p=.395). Therefore, it is acceptable to use the t-test for equality of means to determine whether differences exist between the adoption levels of primary lifecycle practices compared to the other practices.

An independent samples t-test confirmed that the mean of the adoption level of the 13 practices relating to primary lifecycle processes (56.7%; n=13) is significantly higher than that of the combined 31 support/organisation practices (44.1%; n=31) (t=1.76, df=42, p=.043, detailed statistical tables in Appendix F, Tables F.10 & F.11). *Therefore, the null hypothesis for H1b, that there is no variation in the adoption of* *best practices for primary lifecycle processes compared to supporting processes, is rejected.*

5.4.2 Adoption levels by organisation

For each organisation response, the level of adoption of the best practices was calculated. To do this, for each of the 203 valid responses, the number of *yes* responses to the 44 best practice questions were summed and a percentage calculated based on the proportion of *yes* responses to the sum total of *yes* and *no* responses. Blank and *not applicable* responses were ignored in this calculation.

Depending on the calculated level of adoption, respondents were placed in one of four categories to compare the extent of adoption of best practice: up to 25 percent of practices adopted; from 25 to 50 percent; 50 to 75 percent; and more than 75 percent. In §2.7.1, it was hypothesised that respondent organisations would exhibit a wide variation in the adoption levels of best practice techniques. As shown in Table 5.10, adoption levels across the responding organisations vary, with 16 percent of respondents having adopted less than one quarter of the practices, and at the other extreme, almost ten percent claiming to have adopted three quarters of practices.

Table 5.10 Frequencies and percentages of organisations by extent of adoption of practices

Extent of Adoption	Number of Organisations in Adoption Range	% of Organisations
0-24.9%	33	16%
25-49.9%	72	36%
50-74.9%	78	38%
75-100%	20	10%
Total	203	100%

The organisation with the highest rate of adoption achieved 95 percent of the 44 practices (42 yes; 2 no), and the lowest, zero percent (36 no; 0 yes). The mean organisation adoption level of the responses is 47.6 percent with a standard deviation of 21 percent, the large deviation providing further support for the hypothesis that there is wide variation in the level of adoption. The organisation adoption level is slightly less than that reported from the final ESI survey of 1997 (mean 51%; standard deviation 21%) (Dutta, Lee & Van Wassenhove 1999). The variation in adoption levels from the Queensland organisations is even greater than the ESI responses which were described as showing 'wide variation in both awareness and application of process improvement techniques' (Dutta, Lee & Van Wassenhove 1999). *Therefore, the null hypothesis for H1c, that there is no variation in the adoption of software development best practice, is rejected.*

A histogram of adoption levels was produced, and rather than the expected normal distribution, a bi-modal distribution occurred as shown in Figure 5-2.

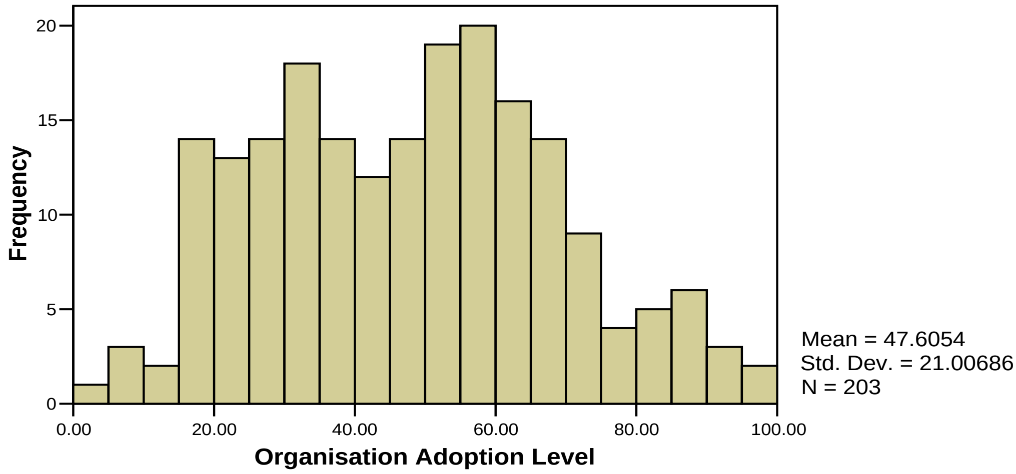


Figure 5-2 Histogram showing distribution of organisation adoption levels.

Assessing normality of *organisation adoption level*. Prior to undertaking any correlation analysis to statistically prove the association between adoption of best practice and variables such as organisation size, the characteristics of the data were explored in order to ensure the correct statistical approach was selected. The assumption of normality is a prerequisite for many inferential statistical techniques. As advised by Coakes and Steed (1996) the K-S (Lilliefors) statistic was calculated to assess if the distribution of organisation adoption level was distributed normally. The significance level of this test on the 203 responses from the survey was less than the required level of .05, indicating that normality cannot be assumed (Kolmogorov-Smirnov (Lilliefors statistic)=.067; df=203; p=.028).

Smaller groups of responses were selected to assess the normality of the adoption level within each group. Firstly, the total number of employees in the organisation was used to group responses with the 129 organisations with less than ten staff selected. This group did not exhibit a normal distribution in regard to the variable organisation adoption level. Next, the 138 respondents with small development teams (of up to five staff) were assessed. This group also failed the test of normality in regard to adoption organisation level. On the basis of the options selected for organisation's primary involvement in the industry, a series of groups from the responses were separated and assessed for normal distribution.

Table 5.11 Results from series of tests of normality on selected groups of responses

Group Selected	K-S (Lilliefors)	df	*p*
All responses	.067	203	.028
Small organisations (<10 staff)	.087	129	.017
Small development team (0-5 staff)	.083	138	.022
Does not develop COTS	.058	116	>.200
COTS Software Developer	.086	87	.152
Custom developers	.099	128	.004
Developers (COTS or Custom)	.089	166	.003
Software Users (in-house or 3rd party)	.076	61	>.200
Not software users (in-house or 3rd party)	.087	142	.011

The results presented in Table 5.11 show that the complimentary groups of non-COTS and COTS developers were the only subsets of responses which exhibit a normal distribution for the variable *organisation adoption level*. Both the non-COTS and COTS developer distributions for adoption are positively skewed, and the negative kurtosis indicates a flatter than a normal distribution (statistical analysis included in Appendix F, Table F.12).

To confirm that the non-COTS and COTS developers represent different populations, the Independent Groups t-test was conducted. There are two assumptions for this test. Firstly, independence of groups: subjects should appear in only one group and these groups are unrelated; and secondly, homogeneity of variance: the groups should come from populations with equal variances. The first assumption was met as the variable COTS was set to zero (indicating the respondent was not a COTS developer) if the COTS option was not selected. To check the second assumption, Levene's test was used to check for equality of variance. This test confirmed that the population variances are relatively equal (Levene statistic based on mean=.493, df=201, p=.483).

Therefore, it is acceptable to use the t-test for equality of means to determine whether COTS/non-COTS differences exist. The t-test indicated that the two groups (non-COTS and COTS) come from different populations (t=-2.73, df=201, *p*=.007). Detailed statistical tests are included in Appendix F, Tables F.13-15.

As the statistical analysis of the distribution of the variable *organisation adoption level* revealed that the responses fall into two distinct populations (COTS and non-COTS developers), subsequent tests involving the variable *organisation adoption level* are conducted separately for COTS and non-COTS organisations.

To determine if organisational characteristics were associated with organisation adoption level, the next four sections (§5.4.3 to §5.4.6) consider the association between adoption level and organisation size, software development group size, development type, and finally industry sector.

5.4.3 Association of adoption level and organisation size

Two questions relating to size were included in the survey. The first question related to the total number of employees in the organisation and the second to the number of employees involved in software development or maintenance.

On the basis of the total number of staff, organisations were scaled as micro-business (less than five staff); small (at least five but less than 20 staff); medium (20 to 199 staff); large (at least 200 and less than 5000 staff), and very large (more than 5000 staff). As can be seen in the summary (Table 5.12), the lowest rate of adoption of best practice was found for micro-businesses (less than 5 staff) for both the non-COTS developers and COTS developers.

Table 5.12 Summary of adoption by organisation size scale

Scale	Non-COTS developers			COTS developers		
	n	%	Mean adoption	*n*	%	Mean adoption
Micro<5	58	50	39.87	33	38	47.87
Small 5-19	21	18	52.42	32	37	50.58
Medium 20-199	15	13	41.87	19	22	58.84
Large 200-4,999	20	17	48.77	2	2	65.03
Very large 5,000+	2	2	53.99	1	1	63.41
Total	116	100	44.18	87	100	52.17

Exploration of the range of the variable *organisation size* revealed three responses with extreme values: organisations with total staff of 30,000, 47,000 and 70,000 employees (as shown graphically in box plot Appendix F, Figure F-1). These three outlier responses were checked to ensure they had been accurately recorded and belonged in the sample (Selvanathan et al. 2000, p. 663). It was predicted in §2.7.1, that organisation size would be positively associated with adoption of best practice. To test the association between the number of employees and adoption level of each organisation, Pearson's correlation test was performed. The association was found to be not significant for non-COTS developers (p=.262) or for COTS developers (p=.292). Statistics are provided in Appendix F Table F.16. *Therefore, the null hypothesis that there is no difference in the adoption levels of best practice for organisations with large or small numbers of staff cannot be rejected and hypothesis H1d, that higher adoption of best practice is associated with the size of the organisation— is not supported.*

However, once the 61 organisations which recorded their primary involvement in the software industry as software *user* (either developed in-house or by a 3rd party) were separated out, a strong relationship was found for the remaining non-COTS (p=.042) and COTS (p=.002) organisations (statistical results in Appendix F Table F.17). An association between organisation size and adoption was not found for the software *user* respondents (for non-COTS: p=.210; for COTS: p=.390 details in Appendix F Table F.18). Therefore, for organisations whose primary involvement in the software industry is not that of software user, adoption of best practice is associated with organisation size.

5.4.4 Association of adoption level and software development group size

Looking at size from the perspective of the number of software developers engaged in programming or maintenance, a stronger pattern emerges. As shown in Table 5.13, organisations with at least 20 developers exhibit much higher average adoption compared to organisations with very small and small development groups.

Table 5.13 Summary of adoption by number of software developers for COTS and Non-COTS

Scale	Non-COTS Developers			COTS Developers		
	n	%	Mean adoption level	*n*	%	Mean adoption level
Micro<5	90	77	41.30%	48	55	46.79%
Small 5-19	10	9	54.61%	32	37	55.39%
Medium 20-199	16	14	53.85%	6	7	72.03%
Large>=200	0	0		1	1	88.64%
Total	116	100	44.18%	87	100	52.17%

To explore the association between adoption levels and size of the software development group, the Pearson product moment correlation coefficient was calculated for the non-COTS development organisations, and also for those organisations which develop COTS systems. A significant positive relationship was found between the organisation adoption level and the number of software developers at each organisation for both non-COTS (p=.003) and COTS groups (p=.002) (Appendix F Table F.19). *Therefore, the null hypothesis, that there is no association between adoption levels and the number of software developers, is rejected and hypothesis H1e—that adoption of best practice is positively associated with the number of software developers— is supported.*

5.4.5 Association of adoption level and type of development

Respondents were asked to indicate the role they played in the software industry, and as detailed in §5.3.5, many provided multiple responses. Table 5.14 lists the mean adoption level for each group.

Table 5.14 Mean adoption by primary involvement in software industry

Primary involvement in industry	N	Mean adoption level
Software user – developed in-house	53	47.00
Software user – developed by a 3rd party	43	41.69
Software developer producing off-the-shelf systems	87	52.17
Software developer producing custom software systems	128	47.94
Research and development institute or university	10	43.42
Interest group e.g. professional society or standards body	2	41.48
Other	7	48.34

To account for the overlap caused by organisations selecting multiple groups, and to focus solely on organisations which develop software, the adoption levels were examined more closely as shown in Table 5.15. In comparing adoption levels of COTS and custom software developers, highest adoption of best practice was reported from organisations which develop COTS software (see Table 5.15).

Table 5.15 Comparison of adoption levels for COTS and custom developers

Developer type	N	Mean adoption level
Develop COTS software	87	52.17%
Do not develop COTS software	116	44.18%
Develop Custom software	128	47.94%
Do not develop Custom software	77	44.18%

To determine if there was a difference in the mean adoption level of organisations which develop COTS software compared to organisations which do not develop COTS software, an independent sample t-test was calculated and a significant difference confirmed (t=-2.726, df=201, p=.007). *Therefore, the null hypothesis H1f, that there is no variation in the extent of adoption of best practice techniques depending on development type is rejected. The extent of adoption is higher in organisations which develop COTS software compared to organisations which do not develop COTS software.*

A similar test was then conducted to compare the best practice adoption of organisations which provide custom development with those who do not provide custom development. As the responses came from two distributions, it was necessary to partition of the responses based firstly on non-COTS/COTS, as summarised in Table 5.16.

Table 5.16 Comparison of mean adoption by type of development

	Develop COTS software		Do not develop COTS	
Custom developer?	*n*	Mean adoption	*n*	Mean adoption
No	37	40.7%	38	53.2%
Yes	79	45.8%	49	51.3%
Total	116	44.3%	87	52.1%

The comparison revealed that there is no significant difference between custom or non-custom developers within either the non-COTS group (t=-1.291, df=114, p=.199, 2 tailed test) or the COTS group (t=.407, df=85, p=.685, 2 tailed test).

5.4.6 Association of adoption level and industry sector

Respondents were asked to select the industry sector to which they belonged. The majority of respondents chose software development (77%). Table F.20 in Appendix F provides a full list of all the industry sectors reported, the number of organisations in each sector, and the mean adoption level of organisations in each sector. To compare the adoption level of organisations within each sector, ANOVA tests were used. As population normality is a prerequisite assumption for analysis of variance, the calculation was performed separately for the non-COTS developers and for the COTS developers. As the second assumption for the ANOVA test is homogeneity of variance, Levene's test was performed in each case.

For the 116 responses from organisations that do not develop COTS systems, Levene's test was not significant (p=.215) providing confidence that the population variances for each group by sector are approximately equal (results in Table F.21 Appendix F). The result of the analysis of variance calculation indicated that the null hypothesis can be rejected as the adoption level varies by sector (p=.037, detailed results in Table F.22 Appendix F). Box plots give a graphical representation of adoption levels of non-COTS developers by sector in Figure F-2 in Appendix F.

As SPSS is unable to perform *post hoc* tests on groups with fewer than two cases, the sectors with only one response were excluded from the multiple comparison. Table 5.17 lists in rank order by adoption the sectors with at least two responses. The analysis of non-COTS organisations (summarised in Appendix F, Table F.23) shows that organisations from the finance and insurance sector, and the utilities sector exhibit significantly higher levels of adoption of best practice compared to organisations from the mining or education sectors. Furthermore, organisations involved in the government administration and defence sector, software development sector or information technology sector all show higher adoption compared to those from the education sector.

Table 5.17 Adoption of non-COTS organisations ranked by sector

Rank order	Sector #	Industry sector	N responses	Mean adoption level	Std dev
1	10	Finance & Insurance	2	62.2%	16%
2	4	Utilities	4	58.2%	12%
3	12	Government Admin & Defence	5	50.0%	7%
4	0	Software Development	78	45.7%	20%
5	3	Manufacturing	3	42.5%	25%
6	18	Info Tech	9	39.1%	16%
7	14	Health & Community Service	2	36.8%	30%
8	9	Communication Services & Media	2	27.6%	11%
9	2	Mining	2	22.3%	7%
10	13	Education	5	16.1%	20%

For the 87 COTS respondents, the same procedure indicated the population variances for each group by sector are approximately equal (Levene's test: p=.179 results in Table F.24 Appendix F); but significant differences did not exist across different sectors (p=.359, detailed results in Table F.25 Appendix F). (The sectors with only one case are excluded from Levene's test but are included in the ANOVA.) However, this result may be unreliable because 90 percent of all the COTS developers belonged to the Software Development sector, leaving few responses in other sectors.

Therefore, the null hypothesis for H1g is rejected for non-COTS developers as the adoption level means by sector are not equal. The adoption level of software development best practice is associated with the industry sector of the development organisation for non-COTS developers but not for COTS developers.

5.4.7 Summary of hypothesis tests

The first research question asked: is there wide variation in the extent of adoption of software development best practice techniques by software developers in Queensland, and is adoption related to particular organisational characteristics? As summarised in Table 5.18, all the hypotheses formulated to answer this question received partial or full support from the data analysis. How these contribute to answering the first research question is discussed fully in Chapter 8.

Table 5.18 Summary of results of hypothesis tests

	Hypothesis	Result
H1a	There is wide variation in the adoption levels of individual best practices	Supported
H1b	Adoption of primary lifecycle practices is higher than supporting and organisation practices	Supported
H1c	There is wide variation in the adoption levels of best practice by organisations	Supported
H1d	Adoption of software development best practice is positively associated with organisational size	Supported for organisations whose primary involvement is not software user
H1e	Adoption level of software development best practice is positively associated with development team size	Supported
H1f	Adoption level of software development best practice varies depending on development type (custom or COTS)	Supported: COTS developers have higher adoption levels compared to non-COTS
H1g	Adoption level of software development best practice varies depending on the industry sector of the development organisation	Supported for non-COTS developers

5.5 Analysis of survey respondents' comments

In Part C of the best practice questionnaire, respondents were asked: *Which aspects of your software development activities have the most scope for improvement?* For this question, 105 of the respondents provided comments. The comments, summarised by theme, are shown in Table 5.19.

Table 5.19 Summary of comments regarding aspects needing improvement

Theme of Comment	Number of Responses
Technical issues: design, testing, configuration management, tools and automating processes	34
Management issues: project management including planning and control, funding/resources, costing, estimating, budgets	29
QA issues including error analysis, standards, documentation, procedures	20
Human resources issues: finding suitable staff, staff development and training, change management, culture	12
Client issues: marketing, client commitment, contract management, requirements analysis and management	7
Most/all aspects!	3
Total number of comments	105

The highest scoring group of comments (by 34 organisations) related to technical issues, specifically relating to testing, and the need for automated support, especially for configuration management. Project management issues were identified as problematic by a large group of respondents (29); and also quality assurance issues (20 responses). At the time of the survey, there was a shortage of skilled development staff, reflected in responses from 12 organisations. Managing clients in general and requirements in particular was mentioned by seven organisations as needing to be improved. These comments highlight processes which organisations identify as needing improvement, in particular processes related to requirements elicitation, software development, configuration management, project management, and quality assurance.

The final item on the questionnaire asked: *'How can SEA help you achieve those improvements?'* There were 77 comments as summarised in Table 5.19. The suggestions indicate many organisations see a requirement for advice and training courses, and view the role of SEA to network software developers by providing contacts.

Table 5.20 Summary of responses: future assistance from SEA

Theme of Comment	Number of Responses
Provide advice/information: estimates; schedules; security; tools; SQA; process documentation; methodologies; templates; updates	26
Don't know, not sure	16
Other services: incubator offices; contacts e.g. beta testers; provide infrastructure and publicity	16
Training courses/seminars: QA; development tools	12
Trivial/irrelevant comment	5
Improve university courses: all aspects of software development; relevance	2
Total number of comments	77

5.6 Summary

This chapter reported the statistical analysis conducted on the survey data. A total of 203 valid responses were received from organisations which develop software. Most of the organisations were SMEs and generally, the software development group comprised less than 10 staff. As far as involvement in the software development industry, almost two-thirds of the respondents develop custom software systems, and a substantial proportion develops commercial off-the-shelf software packages (43%).

The survey indicated Queensland organisations involved in software development have adopted, on average, almost half of the practices listed in the best practice questionnaire. Primary lifecycle practices, such as those relating to requirements and software development, exhibit higher rates of adoption compared to organisation and supporting practices. The adoption of individual practices varies widely, as does the adoption level of organisations. Organisations involved in developing COTS systems have significantly higher adoption of best practice compared to those who not develop COTS systems. The size of the software development group is associated with the adoption level. Also, for organisations not developing COTS systems, higher adoption levels were recorded for organisations from the following sectors: finance and insurance, utilities, government administration and defence, software development and IT. Lower adoption levels were reported from the mining and education sectors.

Half of the respondents identified scope for improvement in their current processes, especially in relation to technical issues such as design, testing, configuration management, tools, and also project management and quality assurance.

The second stage of the study, involving the field experiments, focuses on a smaller number of firms and provides a more detailed analysis of the processes in place and the outcomes of an assessment-based process improvement program (PIP). The next chapters present the field experiment method (Chapter 6) and findings from evaluation of the PIP (Chapter 7). The findings from both stages of the study, the best practice survey and the process improvement program, are discussed in Chapter 8, which also includes a comparison of the survey results with the field experiment results.

6 CHAPTER SIX—FIELD EXPERIMENT DESIGN

6.1 Introduction

In Chapter 3, the research design was described and it was explained that a multi-method approach was used in the study, comprising two stages: firstly, a survey of the software development industry, and secondly, field experiments involving small development firms. The survey instrument design and execution was detailed in Chapter 4, and the data analysis of the survey responses presented in Chapter 5. In this chapter, the design and protocol of the field experiments is described and the analysis of the data obtained from the field experiments is presented in Chapter 7.

Firstly, the field experiments are detailed in terms of the selection of subjects, the design and protocol of the field experiment, and the compilation of results. Then the variables are defined and data analysis method is described. Finally, limitations to the field experiment methodology and ethical considerations are discussed.

6.2 Process improvement program

In §3.5.3, it was explained that the field experiments involved 22 small software development organisations participating in a process improvement program (PIP). The program reported in this thesis was designed and executed by staff and associates of the Software Quality Institute and funded by SEA Queensland. As well as analysing the reports from all the experiments, the author of this thesis participated in three of the 22 field experiments as support assessor and follow-up assessor. The process improvement program entailed an assessment of software development processes at each site (pre-test), recommendations for improvements were made, then 7 to 16 months later, a follow-up meeting was arranged at each site to evaluate improvements (post-test).

Researchers at SQI developed a procedure to enable Rapid Assessments for Process Improvement for software Development (RAPID) (SQI 1999a). The RAPID method was based on the ISO/IEC TR 15504 standard and designed to enable assessments to be performed in one day (Rout et al. 2000; SQI 1999b). The PIP project was managed by one of the members of ISO/IEC JTC1/SC7 Working Group 10, with assistance from the regional SPICE trials coordinator. SQI associates, all of whom had completed the requirement for SPICE Assessor certification, participated in the PIP project. Table 6.1 shows the roles and responsibilities of the PIP members.

Table 6.1 Roles and responsibilities of PIP participants

Role	Responsibilities
PIP manager	Plan project, secure funding, allocate resources, monitor progress.
Regional SPICE trials coordinator	Record assessment data in SPICE Trials Database.
Sponsor	Software development manager or owner of firm. Committed firm to PIP, participated in planning of assessment, provided staff for assessment.
Lead assessor	Plan assessment, prepare report, check details with sponsor, finalise assessment report.
Support assessor	Assist the lead assessor in performing assessment and preparing report.
Follow-up assessor	Plan follow-up meeting, conduct follow-up meeting, prepare and submit final report.

6.3 Selection of subjects

As mentioned in §4.2, the best practice survey gave respondents the opportunity to show interest in participating in software process improvement programs. The positive responses provided a candidate list for the selection of subjects for the process improvement program. The list of candidate firms was reviewed by the PIP manager, who then contacted software development managers, outlined the program and asked if they wished to participate in the process improvement program. The available funding from SEA (Qld) set a quota of 22 firms for the PIP project.

6.4 Design of field experiment instruments

The PIP field experiments used the RAPID assessment method developed by the PIP manager, the SPICE trials coordinator and key staff from the SQI (SQI 1999a, 1999b). The RAPID method comprised a set of six instruments which are described in detail in §6.7.

6.5 Assessment model

Before describing the PIP protocol and assessment instrument, the underlying model used in the PIP experiments is explained. The measurement framework for ISO 15504 uses a two-dimensional reference model of processes and process capability (Rout & Simms 1998).

Process dimension. The ISO 15504 standard sketches out a roadmap for the implementation of best practice in software engineering by defining 40 processes, divided into five categories: customer-supplier (10); engineering (9); support (8); management (4); and organisation (9). The process capability of each defined process 'measures how well each process is managed to achieve its purpose and the organisation's objectives for it' (SPIRE 1998 p.57). Capability is measured in levels from incomplete (level 0), through performed (1), managed (2), established (3) predictable (4) to optimising (level 5) as shown in Table 6.2. These capability levels represent milestones along the road to software process improvement.

Table 6.2 Capability levels

Level	ISO 15504 capability levels	Software process improvement in regional Europe (SPIRE) capability level descriptions
0	Incomplete	Chaos reigns
1	Performed	Do your own thing
2	Managed	Teams rule
3	Established	The organisation learns
4	Predictable	Management by number
5	Optimising	Optimising

The 40 processes defined in ISO 15504 are grouped into five process categories:

- *customer-supplier* processes directly impact on the customer, support the development and transition of the software to the customer, and provide for the correct operation and use of the software product and/or service
- *engineering* processes directly specify, implement, and maintain the software product, its relation to the system, and its customer documentation
- *support* processes may be employed by any of the other processes at various points in the software lifecycle
- *management* processes contain generic practices that may be used by anyone who manages any type of project or process within a software lifecycle
- *organisation* processes establish the business goals of the organisation and develop process, product, and resource assets that, when used by the projects in the organisation, will help the organisation achieve its business goals (ISO/IEC TR 15504-6 1998, p. 4).

Although there are 40 individual processes defined in ISO/IEC TR 15504 (listed in Appendix D Table D.4), organisations are encouraged to select the processes which apply to them, and that can provide the most benefit. As the PIP assessments were restricted to one day each, the scope of the assessment was limited to eight key processes, as listed in Table 6.3.

Table 6.3 RAPID processes and process categories

RAPID Process	RAPID code	Process category	ISO 15504 identification
Requirements elicitation	RE	Customer-Supplier	CUS.3 Requirements elicitation
Software development	SD	Engineering	ENG.1 Development
Configuration management	CM	Support	SUP.2 Configuration management
Quality assurance	QA	Support	SUP.3 Quality assurance
Problem resolution	PR	Support	SUP.8 Problem resolution
Project management	PM	Management	MAN.2 Project management
Risk management	RM	Management	MAN.4 Risk management
Process establishment	PE	Organisation	ORG.2.1 Process establishment

Selection and description of processes. The primary processes of requirements elicitation and software development provide the core of the RAPID model. The PIP manager was particularly interested in the capability of small firms in relation to the processes that support capability level two—*managed* (project management; quality assurance; configuration management; problem resolution); and processes to provide a platform for improvement beyond the managed level to capability level three—*established* (risk management and process establishment) (Rout et al. 2000).

The processes for inclusion in the RAPID method were selected by the PIP manager, the SPICE trials coordinator and key staff from the SQI on the basis of expert judgement. The processes chosen are consistent with areas needing improvement as recorded by respondents to the best practice survey (as summarised §5.5). Also, these processes are considered to be important by other researchers in the field, for example, a survey based on the Software Excellence model included questions relating to a similar set of processes as that selected for the PIP: project management, risk management, requirements elicitation, software development, problem resolution, process establishment, setting standards (quality assurance) and change (configuration) management (Dutta & Van Wassenhove 1997b, p. 11). Furthermore, four of these eight processes (requirements management, project management, configuration management, and quality assurance) were selected to enable the Software Engineering Division at the Sacramento Air Logistics Center to achieve CMM level 2 (Westaway 1995). Process establishment was included as it is important to ensure new processes are institutionalised or developers may go back to using the previous processes (Bamford & Deibler 1998; Butler 1997; Wiegers 1998; Yamamura & Wigle 1997).

Description of PIP processes

The eight processes covered in the RAPID method are now briefly described. For each process, the descriptions are sourced from the ISO 15504 reference model for processes and process capability (ISO/IEC TR 15504-2 1998).

Requirements elicitation. The purpose of the *requirements elicitation* process is to gather, process, and track evolving customer needs and requirements throughout the life of the software product and/or service so as to establish a requirements baseline that serves as the basis for defining the needed software work products. In the RAPID documents this process is also referred to as *requirements gathering.* The management of user requirements has been flagged by many researchers as a critical source of risk (El Emam & Birk 2000a, p. 541; Jeffery 1993; Jones 1994). This process aims to establish a requirements baseline to define the software work products. The assessment team considered whether the firm had established continuing communications with the key customers, and had gained a clear understanding of the customer's requirements for each project. The requirements elicitation process should identify new customer needs, monitor needs on a continuous basis and reflect such needs in the requirements. Another aspect of the requirements elicitation process is to provide customers with the ability to readily establish the status of their requests, and to have a program for ongoing enhancement of the firm's products.

Software development. The purpose of the software development process is to transform a set of requirements into a functional software product or software-based system that meets the customer's stated needs. According to ISO/IEC TR 15504, software development is a high level process made up of seven component processes: system requirements analysis and design; software requirements analysis; software design; software construction; software integration; software testing; and system integration and testing. The RAPID method assessed software development as a whole, without disaggregation into its component processes (Rout et al. 2000, p. 48). The decision to treat software development as the high level process was driven by the time and resource constraint of performing each assessment in one day. The assessment team enquired about the generation of intermediate products such as requirements specifications, design descriptions, integration test reports, and system test reports. The software development process ensures that all of the requirements are addressed in the design, and that the product developed meets the initial requirements. Formal acceptance by the customer is also included in this process.

Configuration management. The configuration management process aims to establish and maintain the integrity of all the work products of a process or project. As well as having a strategy for configuration management, the firm should identify, define and baseline all items generated. Controls for modifications and releases of the products should exist, and the status of the individual products and requests for modification need to be recorded and reported. Modifications and new releases need to be checked for consistency, and the storage, handling and delivery of modifications and new releases of the product controlled.

Quality assurance. The purpose of the quality assurance process is to provide assurance that work products and processes of a project comply with their specified requirements and adhere to their established plans. This process involves the development and implementation of a strategy for performing software quality assurance, and recording and storing evidence of software quality assurance activities. As well as identifying problems or non-conformances with contract requirements, this process enables the firm to demonstrate that the processes and activities conform to relevant standards, procedures and requirements. Note that the creation of the standards and procedures is not part of the quality assurance process, but is in the process establishment process.

Problem resolution. The *problem resolution* process ensures that all discovered problems are analysed and resolved and that trends are recognised. This process ensures that such problems and non-conformances are reported. Furthermore, trends in identified problems need to be recognised so that action can be taken to resolve them.

Project management. The *project management* process identifies, establishes, coordinates and monitors activities, tasks and resources necessary for a project to produce a product and/or service meeting the requirements. This process includes defining the scope of the work, evaluating the feasibility of achieving the goals, estimating tasks and resources, and identifying and monitoring interfaces between the project, and other projects and departments. These activities enable a project plan to be developed and implemented. As well as monitoring and reporting progress against the plan, actions need to be taken to correct deviations and to prevent recurrence of problems identified in the project.

Risk management. The purpose of the risk management process is to identify and mitigate the project risks continuously throughout the lifecycle of a project. The process involves establishing a focus on monitoring of risks at both the project and organisation levels. Therefore, for each project, the scope of risk management needs to be defined and determined, and then an appropriate risk management strategy defined and implemented. As the project plan is developed and monitored, risks to the project should be identified and analysed, and the results used to prioritise resources to monitor these risks. As well as taking actions to correct or avoid the impact of risk, measures of risk need to be defined, and then applied to assess the risk status of projects.

Process establishment. The process establishment process defines a suite of organisational processes for all software lifecycle processes as they apply to the business activities. This process involves the organisation-level approval of a standard set of policies and methods, including descriptions of the common tasks and activities to be followed in every project. The plans and procedures for all projects should be based upon these standards, but can be modified for each project.

This aspect of SPI has been neglected in the past, and was included in the PIP to ensure that the new processes spawned by the PIP, as well as existing processes, are diffused and institutionalised throughout the firms (Iversen et al. 1998, p. 460).

Process capability dimension

The assessment of process capability was limited to the attributes related to capability levels one *performed*, two *managed* and three *established*. Levels four *predictable* and five *optimising* were not included in the RAPID model for two reasons. Firstly, to limit the scope so as to achieve the objective of performing each assessment in one day, and secondly, previous research indicated that it was unlikely that any of the small development firms involved in the PIP would be using processes above level three. In fact, Iversen et al. (1998) restricted their study to CMM level two, as only one third of companies had been assessed at higher than level one. In some of the analysis reports from the SPICE phase 2 trials, data relating to capability levels four and five were excluded 'because of the dearth of observations in levels 4 & 5' (Jung et al. 2001, p. 221).

As shown in Table 6.4, the first level of capability, *performed*, is determined by one attribute *Process Performance*. This attribute is used to determine if the process is performed, at least informally. For the second level, *managed*, there are two process attributes to measure the extent to which the process is performed, and the extent to which its work products are managed. At the third capability level, *established*, two process attributes are examined to determine if the process is defined, and the extent to which suitable resources are provided for the process. Table 6.4 summarises the capability levels, associated attributes and evaluation criteria.

Table 6.4 Process capability levels and corresponding process attributes

Level	Process attribute		Evaluation criteria
1 Performed	PA1.1	Process performance	Is the process performed, at least informally?
2 Managed	PA2.1	Performance management	Is the performance of the process managed?
	PA2.2	Work product management	Are the work products of the process managed?
3 Established	PA3.1	Process definition	Is the process defined?
	PA3.2	Process resource	Are suitable resources provided for the process?

(source: Rout, Hodgen & Tuffley 1999a, pp. 6-7)

In the ISO 15504 model, there are two additional process attributes used to rate the fourth level predictable, how the process is measured and controlled; and a further two for level five optimising: process change and continuous improvement. These four process attributes relating to levels four and five were not included in the RAPID model.

For each of the eight processes examined, the assessors followed the script of the assessment instrument to determine the extent to which the process attributes have been achieved using a four point scale: not achieved, partially achieved, largely achieved and fully achieved. Table 6.5 describes the range and extent of achievement required for each of the four points on the achievement scale.

Table 6.5 Calibration of achievement attribute

Achievement scale	Range of achievement	Description of extent of achievement
N Not achieved	0%-15%	There is little or no evidence of achievement of the defined attribute in the assessed process.
P Partially achieved	16%-50%	There is evidence of a sound systematic approach to and achievement of the defined attribute in the assessed process. Some aspects of achievement may be unpredictable.
L Largely achieved	51%-85%	There is evidence of a sound systematic approach to and significant achievement of the defined attribute in the assessed process. Performance of the process may vary in some areas or work units.
F Fully achieved	86%-100%	There is evidence of a complete and systematic approach to and full achievement of the defined attribute in the assessed process. No significant weaknesses exist across the defined organisational unit.

(source: Rout & Griffith University 1998)

The RAPID method collected evidence only by interview, but participants could illustrate issues under discussion by reference to documents. For each of the eight processes above, questions were asked by the assessors to determine the achievement of the process attributes.

Determination of capability level

The capability level (0, 1, 2, or 3) for each process was determined based on the achievement of the process attributes (as listed in Table 6.4). If the process performance attributes were not achieved, or only partially achieved, the process level is rated at zero—*incomplete*. On the other hand, if the process performance attributes are largely or fully achieved, then the process is rated at level one—*performed*. To gain a rating at higher capability levels, the prior levels must be fully achieved, and the process attributes defined for the current level must be either largely or fully achieved.

In the next section (§6.6), the protocol of the field experiments is described, then in §6.7 the various components of the instrument are detailed.

6.6 Protocol of each field experiment

During the last six months of 1999, 22 organisations accepted the invitation from SQI to participate in the Process Improvement Program. Apart from one firm based in a regional city, all the companies were located in the State capital city. The author of this thesis participated in the program in the role of support assessor for three organisations and assessor for two follow-up meetings.

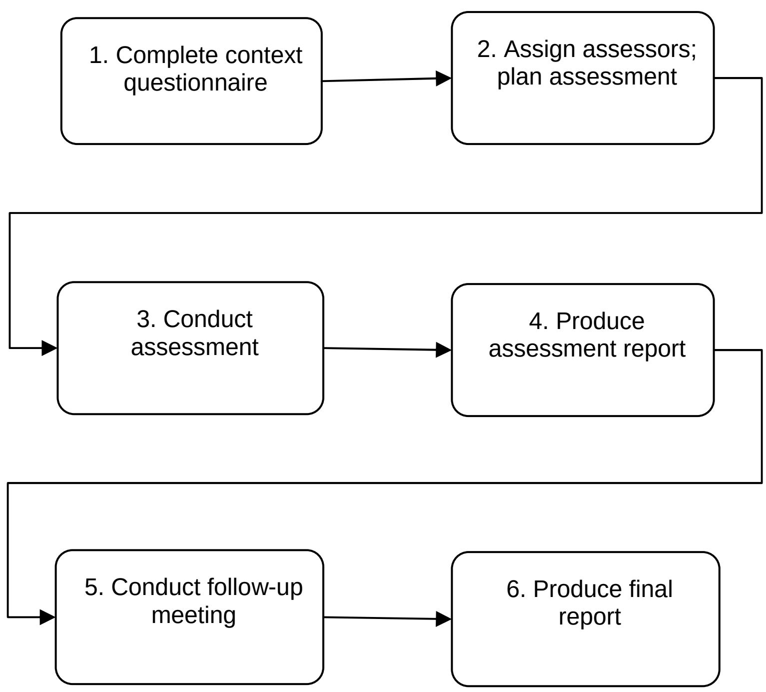


Figure 6-1 Summary of PIP method showing six steps for each assessment

A summary of the six steps for each assessment in the PIP method is given in Figure 6-1. There were three phases for each assessment: preparation phase, assessment phase, and evaluation phase. The steps in the numbered boxes in Figure 6-1 are now discussed.

Preparation phase

Step (1). The organisation context questionnaire was sent to the prospective sponsor for completion and then returned to the PIP manager. The return of this questionnaire signalled the start of the assessment: an assessment plan was then devised, assessors were allocated to the firm, and the assessment was scheduled. A copy of the organisational context questionnaire is included in Appendix G, Section 1.

Step (2). Two trained SPICE assessors were assigned to each firm, one in the role of team leader and the other as support assessor. A set of procedures and templates was provided to members of the assessment team including the completed organisation context questionnaire, the assessment plan, assessment instrument, assessment report template, follow-up interview record and final report template.

Assessment phase

Step (3). The assessment was conducted at the office of the software development organisation over a time period of about six hours. The on-site interviews were conducted by the lead assessor and support assessor with key people involved in managing the software development effort of the firm (usually including the assessment sponsor). For each of the eight processes examined, the assessors followed the script of the assessment instrument to determine the extent to which the process attributes were achieved using the four point scale: not achieved, partially achieved, largely achieved and fully achieved (as presented in Table 6.5). A sample of the assessment template is included in Appendix G, Section 2. Capability indicators for requirements gathering level one are listed in Rout et al. (2000); and the process performance indicators and process capability indicators are published in Rout et al. (1999a).

During the assessment, one assessor asked questions and the other acted as scribe; assessors would alternate these roles. At the end of the assessment, general agreement on the key findings and action items was obtained with the sponsor. The importance of discussing recommendations was stated by Iversen et al. (1998, p. 454); it gives the assessors a more complete picture of the organisation and provides validation to results. Also, discussing the recommendations provides motivation to improve as the assessment participants have a better understanding of the problems.

Step (4). A draft of the assessment report was then prepared by the lead assessor in collaboration with the support assessor and forwarded to the sponsor at the organisation to confirm that the assessment team had accurately recorded the information gathered at the interview. The improvement opportunities were generally expressed in terms of addressing critical risks to the firm's operations or business success. Any changes suggested by the sponsor were further discussed and then the assessment report was submitted to the organisation sponsor, SEA (Qld) and SQI. A feedback form was sent with the assessment report to the sponsor to solicit comments regarding the conduct and value of the assessment.

Evaluation phase

Step (5). Six months after the assessment, contact was made by the assigned follow-up assessor with the sponsor to arrange a half-day follow-up meeting. It was hoped that either the lead or support assessor would be available to conduct the follow-up meeting, but in some cases this was not possible, and another assessor was assigned the responsibility to make contact and arrange a follow-up meeting (the consequences of this are discussed in Chapter 8). At the follow-up meeting, the assessor used the follow-up interview record to review actions taken since the assessment. In some cases, processes were formally reassessed, but where this was deemed unnecessary, a less formal discussion took place.

Step (6). The final report was then compiled by the assessor and submitted to the sponsor and SQI.

The roles and responsibilities of participants in the PIP project are listed in Table 6.1 and more detail is provided in Table 6.6 showing a summary of the allocation of specific tasks to project staff. The actual schedule of assessments and follow-up meetings is provided in Table I.1 in Appendix I.

Table 6.6 Allocation of tasks to PIP project team members

Task	Responsibility
1. Form team of SPICE trained assessors	PIP manager
2. Identify potential participants from best practice survey responses, personal contacts of SQI staff, and contact organisation by phone – are they interested?	PIP manager
3. Send organisational context questionnaire to sponsor	PIP manager
4. Allocate assessment team to assessment organisation	PIP manager
5. Plan assessment, send plan to organisation	Lead assessor
6. Conduct assessment, using assessment templates	Lead assessor and support assessor
7. Prepare assessment report	Lead assessor in consultation with support assessor
8. Send copy of assessment report to sponsor to confirm findings, discuss, refine. Send feedback form to sponsor to comment of value of assessment and assessors.	Lead assessor & sponsor
9. Enter data from Context questionnaire and assessment report into SPICE Trials database	Regional SPICE trials coordinator
10. Allocate assessor to organisation as follow-up assessor	PIP manager
11. Arrange follow-up meeting >6 months	Follow-up assessor
12. Conduct follow-up meeting, using follow-up interview template	Follow-up assessor
13. Prepare final report	Follow-up assessor
14. Send copy of final report to sponsor to confirm details, discuss, refine.	Follow-up assessor

6.7 PIP data collection instruments

The following documents support the RAPID method (Rout, Hodgen & Tuffley 1999b): assessment plan template; organisation context questionnaire; assessment report template; feedback questionnaire; and the process model (described in Rout, Hodgen & Tuffley 1999a). Templates were also provided to the assessors to document the follow-up meetings. The follow-up meetings, although included in the PIP, were not specified in the RAPID method. The templates are now described in terms of how they were used. The variables sourced from these documents are formally defined in §6.9.

6.7.1 Organisation context questionnaire

The organisation questionnaire comprised six parts as detailed in Table 6.7 and provided background information to the assessors about the organisation, employees, and project profile. A copy is included in Appendix G, Section 1.

Table 6.7 Format of organisation context questionnaire

Part	Contents
Part 1	Company data – (fill out form)
	Company name, department, address, contact name, phone, fax and email contact, year founded.
Part 2	Employees – (fill out form)
	Number of employees full time, part time, contract, total
	Education: Postgraduate, graduate, other
	Function: Technical, support/admin
	Experience: more than 5 years, less than 5 years.
Part 3	Market Segment: Application Domain (21 tick box options + other)
	Applied technologies (7 tick box options + other)
Part 4	Projects: Number of projects as Work in Progress, typical no. of employees/project, typical project duration, average cost overrun.
Part 5	ISO 9001: certification (yes/no), date certification awarded.
Part 6	Additional Information: two questions related to performance perceptions.

6.7.2 Assessment instrument

The assessment instrument was developed by the PIP manager in consultation with the regional SPICE trials coordinator and one of the assessors. This instrument was derived from the ISO 15504 standard and used a structured interview questionnaire format. The assessment instrument comprised three parts: a cover page, 49 pages of interview templates and a conclusion section.

The first part served as a cover page and identified the document by recording the assessment number, company details, and names of the assessment team members.

The second part made up the body of the instrument and contained sets of forms to guide the assessor in reviewing the target processes. The requirements elicitation process assessment template forms are included in Appendix G Section 2 as a sample. Each assessment form has three columns: questions, sample artefacts or techniques, then a blank space for the assessor to record interview findings and comments. There were 40 sets of questions, one for each of the five process attributes for each of the eight processes. The questions were not explicitly asked, but rather prompted a moderated discussion and formed guide points for the assessors. The number of questions for each process attribute for each process is shown in Table 6.8. In total, 210 questions were included in the instrument, but all were rarely used as the assessor completes the assessment of each process when it becomes evident that a higher level cannot be attained. The questions relating to PA 1.1 process performance included base practice questions and therefore varied in number and content depending on the process being assessed, however, for the remainder of the process attributes, the same set of generic questions was used, tailored to suit the process under review. Lists of the generic and base practice questions are included in Appendix G, Section 3.

Table 6.8 Number of questions for each process attribute for each process

Process	Process attributes				
	Process performance	Performance management	Work product management	Process definition	Process resource
	PA 1.1	PA 2.1	PA 2.2	PA 3.1	PA 3.2
Requirements elicitation	10	4	4	5	4
Software development	9	4	4	5	4
Configuration management	10	4	4	5	4
Quality assurance	8	4	4	5	4
Problem resolution	7	4	4	5	4
Project management	11	4	4	5	4
Risk management	10	4	4	5	4
Process establishment	9	4	4	5	4
Total number of questions	74	32	32	40	32

The final part of the assessment instrument was the conclusion section. It provided space for the assessor to record action items, agreed process attribute ratings, and a meeting register.

6.7.3 Assessment report

After the assessment had been completed, the lead assessor used the assessment report template to prepare a report of the assessment for the sponsor and SQI. The assessment report comprised five parts:

Part 1: Introduction: report identification, purpose, authorisation, references

Part 2: Background: organisation context, assessment details, summary of findings, capability level ratings

Part 3: Detailed findings: requirements elicitation, software development, configuration management, quality assurance, problem resolution, project management, risk management, process establishment, overall strengths, issues, opportunities for improvement

Part 4: Proposals for action (recommendations)

Part 5: Appendix: assessment record.

6.7.4 Feedback questionnaire

When the assessment report was sent to the sponsor, a feedback form was included for the sponsor to complete and return to the SQI. This provided the opportunity for the sponsor to provide information about the value of the assessment and the competence of the assessors. As these responses do not relate to the variables tested in this study, the details about the feedback questionnaire are not presented here, but are included in Appendix G, Section 4.

6.7.5 Follow-up interview record

The follow-up interview record provided a structured interview format for the assessor conducting the follow-up meeting. It comprised three parts:

Part 1 Assessed company name, contact name, date, follow-up assessor name

Part 2 Agenda items and notes: purpose of meeting. Changes in the external environment: business environment, regulatory requirements, technical infrastructure

Part 3 Review of the assessment report, review of actions taken, relating actions taken to the report recommendations, review capability of target processes, identify and review relevant measures collected. Overall conclusions: value in providing motivation for improvement actions, value in accuracy of the assessment findings.

6.7.6 Final report

The initial assessment report and follow-up interview record were used to compile the final report. The follow-up interview records were retained by the assessors; copies of the final report were sent to the sponsor and SQI. The final report comprised four parts:

Part 1 Introduction: report identification, purpose, authorisation, references

Part 2 Background: assessment details, summary of original findings, overall strengths, risks and opportunities

Part 3 Actions resulting: actions taken, impact on capability of target processes, measures collected, overall conclusions

Part 4 Prioritised action report.

6.8 Compilation of results from field experiments

Prior to commencing the analysis of the field experiments, this researcher located all the relevant documents, checked for missing or ambiguous information (as detailed in §7.3) and then keyed responses from the organisational context questionnaires, the process profiles from the assessment and final reports and the feedback sheets into a MS Excel spreadsheet. Some of the assessment and final reports were provided to the researcher as MS Word documents, the remainder were scanned from hardcopy, or retyped using MS Word. For some of the final reports, the researcher contacted the follow-up assessor to clarify information. The PIP manager was also consulted to explain items on the context questionnaire.

6.9 Definition of PIP variables

The RAPID assessments provided the following variables to be used in the hypothesis tests for the second and third research questions: process capability level and process attribute achievement; organisation and project characteristics.

6.9.1 Process capability level

The assessment report for each firm recorded the *process capability level* for each of the eight processes. Process capability level is an ordinal type variable with a value range from zero (incomplete) to three (established). In published research about the process capability level measurement, many studies treat process capability level as an interval scale measure, for example, analysis of data from phase 2 of the SPICE trials (Jung et al. 2001). Justification for treating process capability level as measured on an interval scale is provided by El Emam and Birk (2000b; 2000a) and Jung et al. (2001, p. 211) who state that the 'coding scheme for process capability lies between ordinal and interval level measurement' as it is a single item measure and is often treated as interval. These researchers refer to many authors who claim that useful studies can be conducted even when proscriptions are violated, for example, 'various studies have shown that, most of the time, using parametric tests for scales that are not strictly interval does not lead, except in extreme departures from the interval scale, to wrong statistical decisions, i.e., one rejects the null hypothesis when one should not' (Briand, El Emam & Morasca 1995, p. 12). However, Lawlis, Flowe and Thordahl (1995) disagree with this approach and assert that 'a combination of descriptive and nonparametric techniques are adequate to establish the presence or absence of a statistically significant correlation of software process maturity and software development success' (p. 7).

Process attribute achievement

There are two criteria which must be met for a process to be rated at a particular capability level: the achievement attributes at that level must be rated *fully* or *largely* achieved; and the attributes at the next lower level must be rated *fully* achieved. It was noted by El Emam (2002) that if only the achieved capability levels are used to measure capability, then the achievement of some attributes at higher levels of capability is ignored. The percentage achievement of each attribute was estimated by the assessor and recorded as ratings according to Table 6.5.

Therefore, to provide finer granularity of measurement, to recognise partial capability within the achievement of each level, and to enable results from this study to be compared with prior research, a measure named *process attribute achievement* was calculated by this researcher for each process for each firm. This was done by firstly applying a numerical coding to the attribute ratings: fully achieved: 3; largely: 2; partially: 1; not achieved: 0; and then summing the numerical attribute codes for each process to produce the attribute achievement score for each process for each firm. In discussing this type of summative model, El Emam and Birk (2000a, p. 554) highlight the fact that some authors consider this method produces interval measurement scales, while others (such as Galletta & Lederer 1989) argue ordinal scales result. Although, as shown in Table 6.5, the intervals between the points on the rating scale are not equal, some prior studies have used parametric statistical tests involving the mean of summed attribute ratings (for example, El Emam 1998, 2002; El Emam & Birk 2000a; Jung et al. 2001). In the statistical tests presented in Chapter 7, process attribute achievement is treated as an ordinal type variable with a value range from zero (no achievement of any of the five process attributes) to 15 (all five process attributes fully achieved).

Other researchers have coded the process attribute achievement ratings by assigning the values 1,2,3,4 to N,P,L,F and referring to this as the 'four-point achievement scale' (El Emam 1998; Jung et al. 2001, p. 211), 'four-point attribute rating scale' (El Emam & Birk 2000a), or the 'process implementation' variable (El Emam & Birk 2000b, p. 130). The author of this thesis decided to use the scale from zero to three as it is more intuitive that attributes judged as not achieved should be coded as zero, rather than one. Considering the sum of the five process attributes for a poorly defined or non-existent process, the 1-4 scale would produce an attribute achievement rating of five, whereas the 0-3 scale would produce an attribute achievement rating of zero. The ratings based on the 0-3 scale can be mapped to the 1-4 scale thereby enabling comparisons of this research with prior studies.

Process capability of primary lifecycle processes

The process dimension of ISO 15504 is based on ISO 12207 (Lifecycle model) illustrated by the groupings of the 24 main processes and 16 sub-processes as listed in Appendix D Table D.4. Two of the eight RAPID processes are classified as primary lifecycle processes: requirements elicitation and software development (ISO/IEC 2002).

Process capability of support, management and organisation processes

The other six RAPID processes represent support (configuration management, quality assurance, problem resolution), management (project management, risk management) and organisation (process establishment) processes.

Extent of process capability improvement

The post-test values for the *process capability level* and *process attribute achievement* for each of the eight processes were derived from the final reports. These variables were calculated in the same manner as the corresponding pre-test values described above from the assessment reports. The extent of process capability level improvement is the difference between the process capability level at the initial assessment and at the follow-up meeting for each of eight processes.

6.9.2 Organisational characteristics

Data for the following variables were derived from the organisation context questionnaire (detailed in §6.7.1). The organisation context questionnaire was sent to the assessment sponsor and returned to SQI prior to the formulation of the assessment plan. As the assessment sponsor was a senior person in the firm, usually the owner or manager of the firm, it is expected that the responses to the questionnaire would be accurate. A copy of the Organisation Context questionnaire is included in Appendix G, Section 1.

Firm size. During 1999, the ABS conducted a review of the way businesses should be defined by size. In summary, the review recommended that for statistical purposes, small businesses (excluding agricultural businesses) should be defined on the basis of full-time equivalent (FTE) employment (2002). Each firm recorded the number of full-time, part-time and contract staff. To estimate staff FTE for each firm, part-time and contract staff were counted as equivalent to half one full-time employee. Firm size is an interval type variable.

Proportion of staff with formal education qualifications. Each firm recorded the number of staff with post graduate qualifications, the number with graduate qualifications, and other. From this, the proportion of post graduate staff members at each firm was calculated by expressing as a percentage the ratio of post graduate staff numbers to the sum of post graduate, graduate and other. In a similar way, the proportion of staff with graduate qualifications was calculated as the number of graduate staff as the sum of post graduate and graduate staff numbers as a proportion of the sum of post graduate, graduate and other. Both formal education proportion variables are interval type with a range from zero to 100. It should be noted that the term graduate proportion includes staff with undergraduate and post graduate qualifications.

Proportion of experienced staff. Firms recorded the number of staff with more than five years experience and the number of staff with less than five years experience. The proportion of experienced staff was calculated as the number of staff with more than five years experience divided by the sum of the number of staff with more than five years and less than five years experience. Staff experience is an interval type variable with a range of zero to 100.

Age of firm. Firms recorded the year founded. From this, 1999, the year of the assessments was subtracted to give the age of the firm in years. Age of firm is a ratio type variable.

Target business sector. Firms were asked to tick, from a range of categories, the industry sector they targeted as clients for their software. The list of industry categories is the same list used in the SPICE trials. The ANZSIC list was not used in this stage of the study due to the need to compare the PIP results with international data from the SPICE trials. This item provided 21 nominal type variables, one for each target business sector with data recorded as *yes* or *no*.

Proportion of technical staff. Firms recorded the number of technical staff and the number of staff in a support or administrative role. No guidance was provided as to the meaning of these terms, it is assumed that sponsors related the terms to the employment award. The proportion of technical staff was calculated as the percentage of technical staff to technical and support/administrative staff. Proportion of technical staff is an interval type variable with a range from zero to 100.

ISO 9001 certification. Firms were asked if they were certified to ISO 9001. ISO 9001 certification is a nominal variable with a value of yes or no. At the time of the assessments, the current standard was ISO 9001:1994, although some firms may have been planning accreditation to ISO 9001:2000.

6.9.3 Project factors

As well as information about the organisation, the sponsor of the assessment provided information on the organisation context questionnaire about the projects undertaken by the firm.

Number of projects in progress. Number of projects currently in progress is a ratio type variable.

Size of project team. Number of staff assigned to each project team. There was no opportunity to record the employment status as full-time, part-time or contract. Size of project team is a ratio type variable.

Project duration. The time duration of a typical project is a ratio type variable. If the sponsor recorded the duration in weeks, it was converted to months.

Project cost overrun. The sponsor at each firm was asked to estimate, as a percentage, the typical proportion over budget for projects. Project cost overrun is an interval type variable.

6.9.4 Perceptions of firm performance

Included in the organisation context questionnaire were two questions to gauge the perceptions of the assessment sponsor in relation to the performance of the firm, such as the ability to meet budget and schedule commitments, and to provide satisfaction to customers and staff. The first question asked for perceptions about performance, the second question asked for perceptions about the importance of a set of performance measures. As well as being part of the SPICE trials data collection suite, these questions were previously used by Herbsleb and Goldenson (1996, p. 325) in a survey of the experiences of firms which had undertaken CMM-based software process improvement.

Herbsleb and Goldenson (1996) asked survey respondents 'to indicate where their organisation stood on several performance measures, including ability to meet schedule and budget commitments, staff morale, customer satisfaction' (p. 325). In the analysis of the data from Phase 2 of the SPICE Trials, the responses to these questions were treated as interval type variables and parametric tests were used to compare perceived performance with process attribute achievement (El Emam 2002, p. 66).

In the organisation context questionnaire, under the heading 'additional information', the first question asked: How would you best characterize your organisation's …

ability to meet budget commitments

ability to meet schedule commitments

ability to achieve high customer satisfaction
ability to meet specified and implied requirements
staff productivity
staff morale/job satisfaction.

Respondents answered the question by marking one tick box on a five point Likert scale: excellent, good, fair, poor, don't know. Following the procedure adopted by Jung et al. (2001), the coding scheme was applied (excellent: 4; good: 3; fair: 2; poor: 1; and don't know: missing value (null)) to provide six ordinal type variables: *perceived budget performance, perceived schedule performance, perceived customer satisfaction performance, perceived requirements performance, perceived productivity performance,* and *perceived morale performance.*

The decision to code *don't know* responses as missing values is based on advice of Rubin, Stern and Vehovar (1995) who assert this is appropriate when there is no intrinsic interest in the fact that a *don't know* response was provided.

The second question related to perceptions about performance and asked how the respondent viewed the importance of the six performance measures. This item was answered by choosing tick boxes on a four point Likert scale: very important; important; somewhat important; not important. The coding scheme was applied (very important: 4; important: 3; somewhat important: 2; not important: 1), providing six ordinal type variables: *perceived importance budget performance, perceived importance schedule performance, perceived importance customer satisfaction performance, perceived importance requirements performance, perceived importance productivity performance,* and *perceived importance morale performance.* The data from this question is summarised in §7.2.3, but is not used in the hypothesis tests. This question was not used in the research reported here for three reasons: it is a very subjective measure and as such does not fit with the positivist approach taken in this study; the question does not directly support any of the three research questions; and finally, it would not be possible to compare the results with prior research as the data from this question has not been reported in any of the research reports from the SPICE trials.

Table 6.9 Summary of variables, scale type and related hypotheses

Variable	Scale	Hypothesis
Process capability level (at assessment) for each of eight processes	Ordinal	H2a-m
Process attribute achievement (at assessment) for each of eight processes	Ordinal	H2a-m
Firm size	Interval	H2b, H3b
Staff experience	Interval	H2c, H3c
Formal education proportion	Interval	H2d, H3d
Proportion of technical staff	Interval	H2e, H3e
Age of firm	Ratio	H2f
Industry sector	Nominal	H2g
ISO 9001 certification	Nominal	H2h
Number of projects in progress	Ratio	H2i
Size of project team	Ratio	H2j
Project duration	Ratio	H2k
Project cost overrun	Interval	H2l
Perceived budget performance	Ordinal	H2m
Perceived schedule performance	Ordinal	H2m
Perceived customer performance	Ordinal	H2m
Perceived requirements performance	Ordinal	H2m
Perceived productivity performance	Ordinal	H2m
Perceived staff morale performance	Ordinal	H2m
Extent of improvement capability level for each of eight processes	Ordinal	H3a-e
Extent of improvement attribute achievement for each of eight processes	Ordinal	H3a-e

6.10 Data analysis method

As noted by Carver (2003), both qualitative and quantitative analysis can be performed on software engineering experiment data. In analysing the PIP field experiments, quantitative methods focused on statistical analysis of numerical data from the context questionnaire and assessment ratings, while qualitative analysis was conducted on the textual content of the assessment and final reports. The use of qualitative techniques with software process research is recommended by Iversen et al. (1998) and Scacchi (1993) to provide opportunities for triangulation and synergy.

In this section, software used for the statistical tests is described, and the statistical tests used for each hypothesis are named and justified. The process attribute achievement ratings and capability levels for each of the eight processes for each firm were checked for accuracy and keyed into MS Excel from the assessment and final reports, as were the responses to the organisational context questionnaires. The highly structured technique of content analysis was applied to transform the data from the assessment and final reports into tabular form (Cnossen 1997), so that lists of assessment characteristics could be compiled such as assessment dates, assessors, and common themes mentioned during the assessments and advice from the assessors. The assessment and final reports were summarised (included in Appendix H).

In the case of 11 of the PIP firms, as a formal assessment was not conducted at the follow-up meeting, qualitative analysis was used to determine if any of the assessment recommendations had been implemented, and if any of the processes had improved. The qualitative analysis also recorded the reasons given for the lack of progress.

The statistical methods used to analyse the PIP data are introduced here, and a glossary of statistical tests and terminology used in this study is included in Appendix B. Prior to undertaking any correlation analysis to statistically prove the association between variables such as firm size and process capability level, the characteristics of the data were explored in order to ensure the correct statistical approach was selected. Although the variables *capability level* and *attribute achievement* are classified as ordinal variables, their distributions were explored. The assumption of normality is a prerequisite for many inferential statistical techniques. As advised by Coakes and Steed (1996) the Kolmogorov-Smirnov Lilliefors statistic was calculated to assess if the values for process capability level and process attribute achievement were distributed normally (test results reported in §7.3).

To determine if a significant difference exists between pre-test and post-test scores for capability level and attribute achievement, the non-parametric pre-test/post-test using the Wilcoxon signed rank test was used (on advice from Bonate 2000).

For ordinal data in the form of related pairs, on advice from Huck, Cormier and Bounds (1974), the Spearman rank order correlation coefficient (rho- r_s) was used to describe the relationship between variables such as the process capability levels and organisation variables. The Mann-Whitney U test was used as the non-parametric alternative to a t-test to study differences between two independent groups for ordinal data (Siegel 1956).

In prior research of process assessments, other researchers have used various statistical techniques. For example, Jung and Hunter (2001) in comparing ISO 9000 certification and size with process capability levels used a non-parametric statistical approach called the bootstrap method to compute the confidence interval, and then the permutation test. To compensate for missing values in the analysis of the SPICE phase 2 trials, multiple imputation was used in some trials analyses (Jung et al. 2001, p. 211). In the field experiments reported here, there were no missing values in the process data as all 22 firms were assessed on all eight processes. For a few of the organisational characteristics, responses were not provided: these missing values were ignored.

Triangulation best practice and PIP data

Data from the best practice survey was triangulated with data collected from the PIP assessments. This was possible as most of the field experiment firms had completed the best practice survey. The attraction of using multiple sources of data is that much more confidence is instilled in the research if the findings are validated through more than one source, and the risks and biases associated with the analysis and interpretation of the data are minimised (Jick 1979; Yin 1994). Spearman rank correlation coefficients were calculated to determine if a relationship exists between the *organisation adoption level* (from the best practice survey) and *process capability levels* (from the PIP).

6.11 Justification for the methodology

As stated in §6.2, the author of this thesis participated in three of the 22 field experiments and analysed the reports from all the experiments. The program reported in this thesis was designed and executed by staff and associates of the SQI and funded by SEA Queensland. The PIP provided a unique and valuable opportunity for the researcher to analyse a large number of assessments. A standard approach was enforced by use of templates, and the PIP procedures were supervised and monitored by the PIP manager.

There were three compelling reasons for the selection of ISO 15504 as the basis for the PIP. Firstly, ISO 15504 is the emerging international standard for process assessment. Secondly, due to the involvement of people from SQI in developing the standard and coordinating the SPICE trials, there was SPICE expertise on hand to manage and monitor the program, and also trained and experienced assessors available to undertake the assessments. Thirdly, the use of the SPICE-based assessment model enabled meaningful comparisons to be made with international SPICE assessments, thereby contributing to the body of knowledge.

6.12 Limitations of the field experiment methodology

In this section, limitations of the methodology are presented and discussed. The discussion is structured based on the four types of validity identified by Cox, Aurum and Jeffery (2004): construct, conclusion, internal and external validity. Validity issues relate to the assessments as well as the evaluation of all the reports.

6.12.1 Construct validity

Construct validity concentrates on the design of the instrument. Flaws in the instrument design can lead to biases in the outcomes (Cox, Aurum & Jeffery 2004). SPICE, the underlying model and approach of the RAPID experiment has been used extensively and validated in phases 1 and 2 of the SPICE trials. In the case of the PIP instruments, all the data collection templates were based on models and templates used in the SPICE phase 2 trials and validated by other researchers.

Although assurance of the reliability of the assessment instrument has been provided, the same cannot be said of the other data collection instrument, i.e. the Organisation Context Questionnaire. Some of the questions were ambiguous and open to inconsistent interpretation by respondents. For example, the assessment sponsor was asked how many staff in a *technical role* but the term technical role is not defined. Is a staff member on the Help Desk counted as a technical or support person? Another example is the item about experience. It is not explained if this refers to years of work experience in business or years of experience in software development. Also, the item relating to education qualifications asked for the number of staff with post graduate qualifications, the number with graduate qualification and *other*. Respondents may have wondered if the qualifications needed to relate to the field of software development, and does *other* refer to other qualifications such as Microsoft certification, or non-computing qualifications, or no qualifications? The purpose of this questionnaire was to provide information to the lead assessor to plan the assessment and to understand the organisation context. The sponsors were encouraged to contact the PIP manager if they required clarification about the context questionnaire, but none availed themselves of this opportunity. Despite the acknowledged problems, the data collected was useful, and instances where questions could be problematic are discussed in the data analysis in the next chapter.

Guidelines for increasing the validity of software process assessment methods have been developed (by El Emam & Goldenson 1998) and were applied in the PIP field experiments. Firstly, all assessors had completed a SPICE certification course and had extensive software development experience. Secondly, a well-documented protocol was followed by the assessment teams and the PIP manager provided guidance to the assessors. Thirdly, the assessment teams were arranged to partner a novice assessor with an experienced assessor. Finally, the feedback questionnaire (described in §6.7.4), improved the reliability of the assessments as it enabled the PIP manager to check the competence of the assessors.

The feedback questionnaires were sent to the sponsor with the assessment report. As well as gathering feedback about the perceived value and accuracy of the assessment, the feedback questionnaire asked about the behaviour and competence of the assessors. Although the sponsors were requested to return the feedback forms to SQI, a full set of forms was not available for analysis. It appears that some forms were either not returned or misfiled. However, from the returned forms, the feedback indicated that as well as demonstrating an understanding of the processes involved, the assessors also displayed an adequate understanding of the business. The assessors acted professionally, did not exhibit bias, and the sponsors felt assured that confidentiality would be not be breached. Most firms reported that they understood the assessment process, the purpose of the assessment activities, its results and the process profile. All of the sponsors who responded to this questionnaire were of the opinion that the assessment was worth the expense and time involved. Generally, sponsors agreed that the assessment helped the firms to understand areas needing improvement, and also that it provided valuable direction about priorities for process improvement. Half the respondents were of the opinion that they needed more guidance about how to go about process improvement. There was general agreement that the correct staff were interviewed, and that they were honest with the assessment team.

In the PIP field experiments, as recommended by Yin (1994, p. 33), a chain of evidence was established by ensuring key informants at each firm reviewed the draft assessment and final reports. This enhances the accuracy of the data in each report. A summary of each of the 22 field experiments is included in Appendix H. All the assessment reports were checked for completeness and accuracy. Also, interpretations of the content of the reports and context questionnaires were checked with the PIP manager. Furthermore, the PIP manager provided clarification to the researcher about the planning and execution of the PIP field experiments. Unfortunately, some of the feedback questionnaires were not available to the researcher as they had been misplaced or not returned from the sponsors.

6.12.2 Conclusion validity

According to Cox, Aurum and Jeffery (2004), 'conclusion validity considers whether the conclusions drawn from the statistical results are valid or are biased by the issues affecting the treatment and the outcome' (p. 225). To ensure that the underlying assumptions of statistical tests were not violated, particular attention was given to the classification of the scale types for capability level and attribute achievement. Furthermore, the distributions of these variables were explored to determine if they conformed to the normal distribution. Despite the fact that other researchers had used parametric tests, this researcher decided that non-parametric tests were appropriate. As well as providing justification and references for the different statistical tests employed, the confidence level of each statistical test is reported.

6.12.3 Internal validity and reliability

Internal validity refers to '... the degree to which the results of the study can be relied upon as being correct' (Davis & Cosenza 1988, p. 114). In relation to the PIP field experiments, internal validity refers to whether the experimental treatment was the sole cause of observed changes in the dependent variable, or were the results influenced by the confounding effects of extraneous variables (Zikmund 1994, p. 304). Zikmund's classification of internal validity threats is now used to examine the PIP field experiments.

History effects may have occurred due to specific events which happened in the external environment between the first and second measurements. Between the assessments in 1999 and the follow-up meetings in 2000, two major events did in fact occur: the year 2000; and the Australian Government's Goods and Services Taxation (GST) legislation came into effect. For many of the PIP firms, both events required an assessment of the potential impact on the software at their installed client base and performance of maintenance activities. For other firms, these events were a source of increased business in the form of new clients, or existing clients wishing to upgrade software. As a result, many firms may have given a lower priority to the implementation of the PIP recommendations due to the timing of the program.

Maturation effects may have resulted due to changes in the development firms over the time period from the assessment to the follow-up meeting. During this time, it is possible that developers, either by self-study or by attending training courses, increased their expertise in their field. Therefore some of the measured improvements could have resulted from the maturation of individual developers, rather than as a result of the PIP intervention.

Testing effects may have occurred if the subjects responded differently to the post-test as a result of the pre-test experience. In the PIP intervention, the developers may not have been aware of the type of information sought by the assessors, but then after reading the assessment report, the developers had a clearer concept of process capability when it came to the follow-up meeting.

Instrumentation effects could have resulted from a change in the procedures or interviewers. There were nine assessors involved in conducting the 22 PIP field experiments; this could have caused extraneous variation as different assessors may have had different standards in measuring the process attributes. The risk of this effect was minimised in three ways: all assessors had undertaken the same SPICE training course; the detailed templates prescribed the procedures; and assessors worked in pairs.

There is no *selection effect* as sample bias did not result from differential selection of respondents for the comparison groups. As the process capability comparison was of a pre-test/post-test type, firms were compared with themselves, not against a control group.

Mortality effect caused by subjects withdrawing from the experiment may have occurred in the PIP field experiments. One firm cancelled its membership of SEA and was therefore dropped from the study prior to the follow-up meeting. Another firm went out of business and could not be traced at the time of the scheduled follow-up meeting.

In summary, the threats to internal validity were minimised as much as possible, with the exception of the inconvenient timing of the program which may have inhibited the effort of some firms to implement the recommendations.

6.12.4 External validity

Although the PIP field experiments involved only 22 firms, there was a reasonable degree of heterogeneity, enabling the results to be generalised to the wider population of small software development organisations. Although all small, these 22 firms targeted different business sectors and produced different types of products (COTS, custom). The group included firms located in the city, suburbs and region, and these firms serviced local, state, national and international clients. Furthermore, at least one assessment was conducted with an in-house software development group making software to be embedded into a product. Therefore, based on the wide variety as detailed above, the group of 22 firms was representative of small firms in Queensland and the results should be generalisable to the broader population of Australian small software development firms.

In summary, while a conscientious attempt was made to ensure that the data collected were as accurate and reliable as possible, a number of limitations have been identified. Constraints imposed by time and cost precluded a full resolution of these limitations. Further limitations and recommendations for their minimisation are discussed in Chapter 9.

6.13 Ethical considerations related to PIP field experiments

Each person involved in the research has certain roles and responsibilities: ethical issues are an important consideration relating to the 'proper conduct of the research process' (Davis & Cosenza 1988, p. 456). In relation to empirical studies of software engineering, researchers, sponsors and potential subjects may all have different reasons to be concerned about ethical issues (Singer & Vinson 2002). If ethical principles, such as informed consent, scientific value, beneficence and confidentiality are not followed, researchers risk losing access to the subjects, to funding, or to other resources (Singer & Vinson 2002).

In this research, the following steps were taken to address ethical concerns. Firstly, prior to commencing the assessments, the PIP manager assured the assessment sponsors that the information provided by staff or managers at the firms would be totally confidential, and that any research published based on the assessment reports would not be traceable back to an individual or organisation that participated in the PIP. Furthermore, informed consent was gained from all participants, and all developers and managers were given the right to withdraw from the assessment or the program. Secondly, all people involved in conducting and recording the assessments (SQI associates and SEA staff) were made aware of the confidential nature of the assessment information. Thirdly, the findings from each assessment were discussed with the participants, and each firm was provided with an individual assessment report.

In analysing and reporting the outcomes from the 22 firms, care was taken in summarising the assessments and final reports to ensure that firms could not be identified. Any specific information which could identify a firm was reworded to a general expression. In the summaries of the assessment reports, care was taken to delete references to the name of the firm's software product, locations, and names of staff. Also, the identity of the assessors was not revealed.

Ethical concerns relating to scientific value are discussed in §6.12 in terms of the accuracy of the data and the use of appropriate statistical tests. In discussing beneficence, the aim is to maximise the benefits and minimise the risks for each stakeholder (Singer & Vinson 2002). It is recognised that within each firm, the assessments may have caused some disruption to staff work activities, and added to stress. Furthermore, in implementing the recommendations, some of the development staff may have experienced feelings of anxiety related to changed processes and new software.

In summary, adequate procedures were followed to ensure the PIP assessments and evaluations were conducted and reported ethically.

6.14 Summary

In this chapter, the research method for the PIP field experiments was described and justified. The variables used in the field experiments were defined. Issues relating to validity, reliability and ethics were addressed. The next chapter presents the analysis of the data collected from the PIP field experiments

.

7 CHAPTER SEVEN—FIELD EXPERIMENT RESULTS

7.1 Introduction

The previous chapter described the method used for the PIP field experiments and defined the variables relating to research questions 2 and 3. This chapter focuses on the results of the field experiments, reporting the findings from the 22 assessments and also the follow-up meetings. Firstly, the 22 firms which participated in the process improvement program are summarised in terms of their time period of operation, application domain, staff formal education, staff experience, and number of staff. Then statistical tests are presented exploring the relationship between the process capability levels and demographics such as organisation size and education level. The extent of process improvement from the time of the initial assessment to the follow-up meeting is analysed. Following this, the data from the survey is compared with that gathered through the field experiments. Finally, a summary of all the results of the hypothesis tests for the three research questions is provided. The interpretation and implications of the findings presented in this chapter are discussed fully in the next chapter.

7.2 Description of firms

In Appendix H, a summary report for each of the 22 firms is provided. Each report summarises the strengths and weaknesses noted at the initial assessment and the events and improvements reported at the follow-up meeting. To retain consistency with the source records, the 22 cases are numbered from #1 to #25. Information relating to firms #6, #10, #20 is not provided as these firms were involved in a different program outside the scope of the RAPID-based PIP field experiments.

As detailed in §6.4, prior to the initial assessment, the sponsor at each organisation completed an organisation context questionnaire. The data collected from this questionnaire were keyed into a MS Excel spreadsheet and analysed using Excel and SPSS to provide a demographic description. The information related to organisational characteristics is reported in §7.2.1, the project-related information is presented in §7.2.2, and the performance perceptions in §7.2.3.

7.2.1 Organisational characteristics of firms

In this section, data collected from the organisation context questionnaire are summarised to provide a profile of the firms included in the process improvement program. A list of all the data relating to the organisation characteristics is provided in Appendix I, Table I.2.

Age of firm. As can be seen from Table 7.1, five of the 18 firms who responded to this question were relatively young firms, less than five years old at the time of the initial assessment. The oldest firm had been operating for 15 years, and four other firms at least 10 years. Nine of the firms had been in business for 5 to 10 years.

Table 7.1 Summary of age of firms

Year founded	Age in years	Number of firms	% of firms	Cumulative %
1984-1989	10-15	4	18%	18%
1990-1994	5-10	9	41%	59%
1995-1998	1-5	5	23%	82%
Did not provide information		4	18%	100%
Total		22	100%	

Organisation size. Many of the firms indicated that contract and part-time staff were involved in their business. The total full-time equivalent (FTE) was calculated (as defined in §6.9.2). Most of the firms were small: 18 of the 22 firms had a staff headcount of less than 50, the other three between 50 and 60 staff (as shown in Table 7.2).

Table 7.2 Size of firms – staff headcount

Staff headcount (FTE)	Number of firms	% of firms	Cumulative % of firms
<5 persons	4	18%	18%
5-10 persons	9	41%	59%
10-20 persons	6	27%	86%
55-65 persons	3	14%	100%
Total	22	100%	

Proportion of staff with formal education qualifications. Overall, the level of formal education of staff employed in the 22 firms was high. Two of the larger firms had very high numbers of post graduate staff, for example, firm #22 reported 70 staff with post graduate qualifications and firm #8 reported 50. When the proportion of staff with post graduate qualifications was examined for each firm, the distribution is far from even, as shown in Table 7.3: ten of the 22 firms do not have any post graduate qualified staff.

Table 7.3 Post graduate qualifications

Proportion of staff with post graduate qualifications (at each firm)	Number of firms	% of firms	Cumulative %
0%	10	45%	45%
10-25%	7	32%	77%
26-50%	2	9%	86%
51-75%	2	9%	95%
100%—all staff have post graduate qualifications	1	5%	100%
Total	22	100%	

The proportion of graduate staff (staff without post graduate qualifications) was quite high. As shown in Table 7.4, 14 of the 22 firms reported that at least half their staff had graduate qualifications.

Table 7.4 Graduate qualifications (excluding post graduates)

Proportion of staff with graduate qualifications (not post graduates)	Number of firms	% of firms	Cumulative %
0-25%	3	14%	14%
26-50%	5	23%	37%
51-75%	5	23%	60%
76-100%	9	41%	100%
Total	22	100%	

To further explore the education issue, post graduate and graduate numbers were summed to calculate the proportion of total staff with university qualifications. As shown in Table 7.5, this calculation revealed that in 12 of the 22 firms, all staff held university qualifications, and only 4 of the 22 firms had less than half of their staff with formal qualifications. The firm with the lowest level of formally qualified staff reported that seven of the nine staff did not have university qualifications.

Table 7.5 Formal education qualifications—graduate and post graduate combined

Proportion of staff with graduate or post graduate qualifications	Number of firms	% of firms	Cumulative %
22-50%	4	18%	18%
51-75%	2	9%	27%
75-100%	16	73%	100%
Total	22	100%	

Proportion of technical staff. Each firm was asked how many staff had a technical role, and how many a support or administrative role in the organisation. To compare the proportion of technical staff to total staff, the percentage of technical staff was calculated: the results are shown in Table 7.6. The mean proportion of technical staff was 71 percent of staff, and at 11 of the firms, at least 75 percent of staff were in a technical role.

Table 7.6 Comparison of proportion of technical staff

Proportion of technical staff compared to total staff	Number of firms	% of firms	Cumulative %
0-25%	0	0%	0%
26-50%	5	23%	23%
51-75%	6	27%	50%
76-100%	11	50%	100%
Total	22	100%	

Experience of employed staff. In response to the question about the number of staff with more than five years experience, 15 of the 21 firms stated that the majority of their staff had in excess of five years industry experience. In contrast, as shown in Table 7.7, two firms reported that less than one quarter of their staff had more than five years experience.

Table 7.7 Comparison of proportion of experienced staff

Proportion of staff with at least 5 years experience	Number of firms	% of firms	Cumulative %
0-25%	2	9%	9%
26-50%	4	18%	27%
51-75%	8	36%	63%
76-100%	7	32%	95%
No response	1	5%	100%
Total	22	100%	

Target business sector. Sponsors were asked to identify which business sectors acquire their software. Eight of the firms chose the ‘other’ response and recorded a sector not listed on the questionnaire. These eight responses were checked against the firms’ web sites. Six were recoded into one of the categories provided, and two new categories were added: mining and agriculture. Six firms focussed their efforts on just one business sector, while the rest developed software for a broader range of industries (as shown in Table 7.8).

Table 7.8 Number of target business sectors selected

Number of target business sectors selected	Number of firms	% of firms	Cumulative %
1	6	27%	27%
2-5	11	50%	77%
6-10	4	18%	95%
Selected all 21 listed sectors	1	5%	100%
Total	22	100%	

The frequencies of responses for each target business sector are listed in full in Appendix I, Table I.3, and the recoding of the 'other' responses is shown in Appendix I, Table I.4. Due to the large number of sector categories, a summary grouping similar sectors was compiled and is presented as Table 7.9. Results show that over half of the 22 firms (55%) provide software to the public utilities and public administration sectors, followed by information technology and software (41%); manufacturing, automotive and distribution/logistics (36%); and telecommunications and media (36%) sectors.

Table 7.9 Distribution of target business sectors

Target business sector group	N	% of Firms
Public utilities and public administration	12	55%
Information technology, software	9	41%
Manufacturing, automotive, distribution/logistics	8	36%
Telecommunications, media	8	36%
Construction and mining, petroleum, agriculture	7	32%
Finance, insurance, banking	6	27%
Leisure and tourism, travel	6	27%
Consumer goods and retail	4	18%
Defence, aerospace	4	18%
Education, health, pharmaceutical	3	14%

ISO 9001 certification. Two of the firms in this study of 22 responded that they had ISO 9001 certification, and provided the date of certification. One firm responded that their certification process was in progress, and another firm based its quality management system on ISO 9001 and held second-party certification to the Queensland Government Quality Assessment Unit (now defunct) standard.

Summary of organisation characteristics. As shown in Table 7.10, most of the firms had been established at least five years, and were staffed by high proportions of experienced and formally educated staff, mainly in a technical role. This group of firms service clients from a broad range of business sectors, and most did not hold ISO 9001 certification.

Table 7.10 Summary of organisational characteristics

Organisational characteristics	N	Min	Max	Mean	SD
Age of firm	18	1 year	15 years	7 years	4.3 years
Staff headcount FTE	22	2	65	15.6	4.0
Education level: proportion of qualified staff*	22	22%	100%	85%	23%
Proportion of technical staff	22	33%	100%	71%	19%
Proportion of experienced staff	21	17%	100%	64%	25%
Target business sectors selected	88	1	21		
ISO 9001 Certification	22	2 yes	20 no	-----	-----
*Note: Number of graduates + post graduates as a proportion of total staff					

7.2.2 Project characteristics of firms

Sponsors were asked to provide information about the number of projects in progress, the staffing level per project, project durations and the cost overrun. A list of all the response data relating to the organisation characteristics is provided in Appendix I, Table I.5. Some of the sponsors responded with a range of values; in these cases, the range mid-point was used for the statistical analysis (for example 5-10 months was recoded as 7.5 months). One firm did not provide any responses to the four questions relating to projects.

Number of projects in progress. The 21 firms were working on a total of 114 projects with the firm average being 5.5 projects per firm in progress. The extreme case was one firm with only 12 full time staff and one contractor claimed to have 34 projects in progress. On the other hand, 16 of the 21 respondent firms reported their business had between one and five projects in progress.

Typical number of employees per project. Considering the number of small firms in the study, it was expected that most project teams were small: the mean number of staff per project was 3.3 staff. Eight firms reported that they typically have two staff per project and four firms have five. Only two of the 21 firms had more than five staff per project.

Typical duration to final delivery. Responses to the item about the typical duration of projects to final delivery ranged from 'less than two weeks' to '5 years so far'. The mean duration was 10 months, and 14 of the 21 firms reported project duration of typically less than six months.

Average cost overrun. Only 13 of the 22 firms provided useable responses to the question about the average cost overrun. Responses ranged from zero to a maximum of 100 percent. The mean was calculated as 23 percent cost overrun.

Summary of project characteristics. The project characteristics in terms of the minimums, maxima, mean and standard deviation for each of the four characteristics are summarised in Table 7.11. Generally, some firms were focussed on few projects while others had a large number in progress, and projects were staffed with about three staff per project.

Table 7.11 Summary of project characteristics

Project characteristics	N	Min	Max	Mean	Std Dev
No. of projects in progress	21	1.0	35	5.45	7.13
Number of employees per project	21	0.5	10	3.29	0.50
Duration of project to final delivery (months)	21	0.3	60	10.40	2.92
Average cost overrun as a percentage	13	0.0%	100%	23%	8%

7.2.3 Perceptions about performance

Included in the organisation context questionnaire were two questions related to perceptions of performance: firstly the perceptions about the firm's ability to meet performance indicators such as budget, schedule and requirements; and secondly, the importance of these performance measures. A Likert scale was used to record agreement to a set of six items. The responses, as summarised in Table 7.12, indicate most firms perceive their ability to meet budget commitments to be in the range of *fair to good*; schedule commitments—*fair*; customer satisfaction—*good*; requirements—*good*; and half the firms perceive staff productivity and morale to be *good*.

Table 7.12 Summary of responses of performance perceptions

How would you best characterize your organisation's …	N	Excellent	Good	Fair	Poor	Don't know
Ability to meet budget commitments	19	4%	41%	32%	9%	14%
Ability to meet schedule commitments	22	9%	23%	54%	14%	0%
Ability to achieve high customer satisfaction	20	14%	64%	14%	0%	9%
Ability to meet specified and implied requirements	22	9%	77%	14%	0%	0%
Staff productivity	22	14%	50%	23%	4%	9%
Staff morale/job satisfaction	21	23%	50%	23%	0%	4%

The sponsor of each assessment was also asked to rate the importance of the series of performance measures. Responses, as summarised in Table 7.13, indicate that most sponsors perceived all measures to be important or very important to their organisation. It is interesting to note that one firm responded that while customer satisfaction and requirements were very important, the other four performance indicators were not important.

Table 7.13 Summary of responses of importance of performance measures

How important are the following performance measures to your organisation:	N	Very important	Important	Somewhat important	Not important
Ability to meet budget commitments	22	50%	32%	9%	9%
Ability to meet schedule commitments	22	45%	45%	4%	4%
Ability to achieve high customer satisfaction	22	73%	27%	0%	0%
Ability to meet specified and implied requirements	22	64%	36%	0%	0%
Staff productivity	22	45%	45%	4%	4%
Staff morale/job satisfaction	22	64%	23%	9%	4%

7.2.4 Summary of demographic profile

To summarise the organisational and project characteristics, most of the group of 22 were small firms with an average staff headcount of about 16 staff, educated to graduate level, with experienced staff in a mainly technical role, targeting a wide range of application domains, and without ISO 9001 certification. On average, each firm was currently undertaking about five projects, with a duration of 10 months, staffed with a team of three people.

The firms generally perceive that they perform better in terms of achieving satisfaction for customers, meeting requirements, and providing job satisfaction for staff, compared to their performance regarding budget, schedule and productivity.

In the next section, the organisation and project characteristics are compared with the process capability levels recorded in the assessment reports.

7.3 Process capability assessments

In this section, the hypotheses relating to the second research question are tested. This section seeks to answer the question: is the capability of technical processes higher than that of management processes, and is process capability associated with particular organisational characteristics? In §2.7.2, the directional research hypotheses are stated, but in each case, the null form is tested in this chapter.

Data preparation. The 22 process profile tables in the assessment reports were checked to confirm that the assessors correctly determined the capability levels based on the process attributes. In two of the 176 processes assessed, the capability level awarded was inconsistent with the attributes listed in the process profile. These two cases were discussed with the PIP manager and the capability levels were changed as it was deemed that the attribute ratings were likely to be correct and that the assessors had mistakenly calculated or recorded the incorrect capability level. The attribute achievement for each process was calculated by applying a numerical coding to the attribute ratings: fully achieved: 3; largely: 2; partially: 1; not achieved: 0; and then summing the numerical attribute codes for each process. As detailed in §6.5, there were five process attributes, therefore each process at each firm could have an attribute achievement in the range from 0 (all process attributes no achieved) to 15 (all process attributes fully achieved).

Capability level and attribute achievement measures. Prior to describing the results of the 22 field experiments, the measures for capability level and attribute achievement are now explored. In Appendix I, Table I.6, the capability levels and process attribute achievement ratings are provided for each process for each firm.

As detailed in §6.9.1, there are two criteria which must be met for a process to be rated at a particular capability level: the attributes at that level must be rated fully or largely achieved; and the attributes at the next lower level must be rated fully achieved. Following Hunter (1998), the 176 process ratings were examined to determine if any would have been rated at a higher level if the second criteria was relaxed to allow attributes at the next lower level to be rated largely or fully achieved. The result, involving 15 process instances as shown in Table 7.14, indicates that five processes would have been rated at level 2 rather than level 1; six processes would have been rated at level 3 rather than level 2; and four processes would have been rated two levels higher at level 3 rather than level 1. Thus in more than eight percent of all process ratings, processes failed to achieve a particular capability level because of inadequacies at the previous level, rather than at the level in question.

While performing this analysis, an extreme example of this case was identified (highlighted in Table 7.14 by *). At firm #8, the achievement attributes for problem resolution did not fully satisfy the level 1 criteria, and yet fully satisfied level 2.

Table 7.14 List of process rating anomalies

Firm #	Process		Attribute rating level 1 performed	Attribute rating level 2 managed	Attribute rating level 3 established	Process capability level	Capability level if prior criteria relaxed	Attribute achievement
4	ENG.1	SD	F	FL	LF	2	3	13
4	SUP.3	QA	F	FL	LF	2	3	13
7	SUP.2	CM	F	LF	LL	2	3	12
7	SUP.3	QA	L	LL	LL	1	3	10
7	SUP.8	PR	L	LL	LL	1	3	10
8	SUP.8	PR	L	FF	LL	1*	3	12
8	ORG.2.1	PE	L	FL	LL	1	3	11
9	CUS.3	RE	L	LL	PP	1	2	8
9	MAN.2	PM	F	LF	LL	2	3	12
11	CUS.3	RE	F	FL	LL	2	3	12
16	MAN.2	PM	L	LL	NN	1	2	6
18	SUP.3	QA	F	LL	LL	2	3	11
19	ENG.1	SD	L	LL	NL	1	2	8
21	ORG.2.1	PE	L	LL	PL	1	2	9
25	ENG.1	SD	L	FL	PL	1	2	10

Note: * indicates a process which fully satisfied capability level 2 but did not fully meet level 1 requirements.

Legend

Processes:
RE Requirements elicitation
SD Software development
CM Configuration management
QA Quality assurance
PR Problem resolution
PM Project management
RM Risk management
PE Process establishment

Attribute ratings:
N Not achieved
P Partially achieved
L Largely achieved
F Fully achieved

Tests of normality. Many of the reports of the SPICE trials data assumed that the variables for capability level and attribute achievement were interval scale measures, and used parametric tests (El Emam & Birk 2000b, 2000a; Jung et al. 2001). In §6.9.1 it was stated that capability level and attribute achievement are ordinal variables, although treated by some researchers as interval. Tests of normality were carried out to explore the distributions and ensure that the correct statistical tests (parametric or non-parametric) were used. Normality tests were carried out on the variables measuring the eight process capability levels, and the eight attribute achievements for the data recorded at the initial assessment and also at the follow-up meetings. According to Coakes and Steed (1996), if the sample size is less than 50, use of the Shapiro-Wilks statistic (rather than K-S Lilliefors) is recommended to assess normal distribution. Analysis of the 32 variables, summarised in Appendix I, Table I.7 shows that only four of the variables exhibited a normal distribution: attribute achievement of requirements elicitation, software development and configuration management at initial assessment, and attribute achievement of requirements elicitation at the follow-up meeting. Therefore, as the variables *capability level* and *attribute achievement* are ordinal type and in most instances are not normally distributed, non-parametric tests were used in the statistical analyses.

7.3.1 Process capability at initial assessment

In this section, the results from the 22 assessments are summarised by capability level, as well as the attribute achievement. Following this, correlations are reported which tested the relationship between organisation and project characteristics (such as firm size, experience of staff, formal education of staff) and process capability levels as determined at the assessment.

A total of 176 process ratings were recorded during the initial assessments, eight processes for each of the 22 firms. As shown in Figure 7-1, most of the processes were rated at level 0 incomplete (30%) and level 1 performed (46%). The frequencies of levels for each process are listed in Appendix I, Table I.8

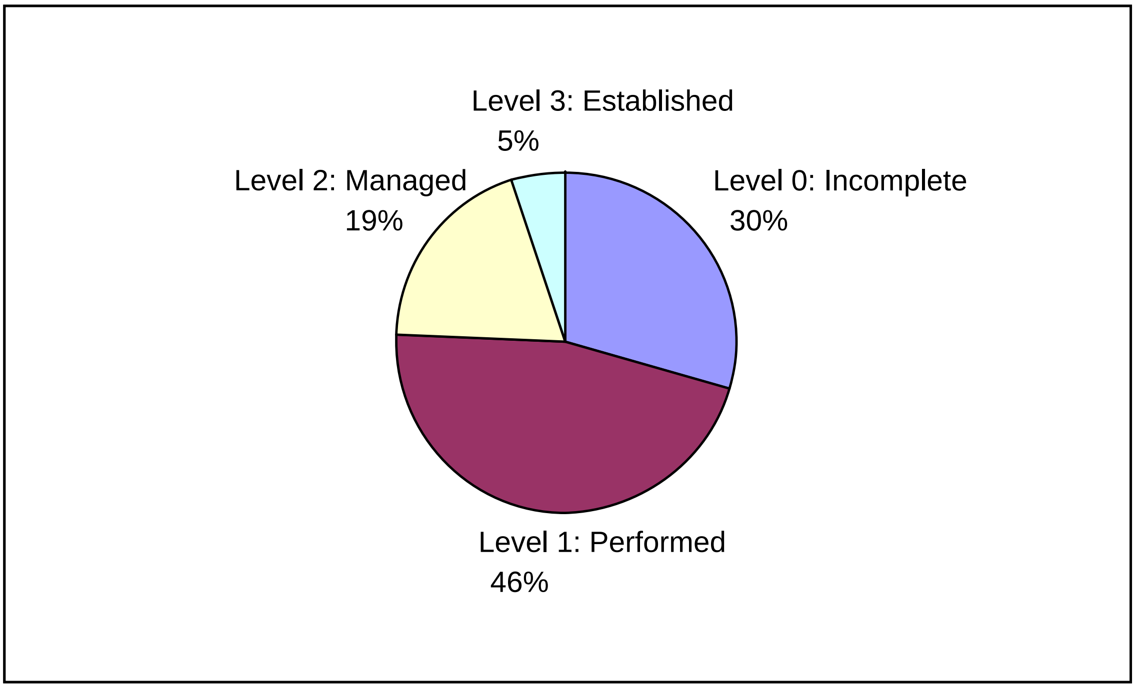


Figure 7-1 Capability level distribution

Furthermore, analysis of the achievement attributes shows, as expected with the implementation of a maturation stage model, attributes corresponding to the higher capability levels (2 and 3) receive less *fully* and *largely* achieved ratings compared to those at the lower levels (0 and 1). The proportion of attribute achievement for each of the five process attributes is presented graphically in Figure 7-2. Consistent with the preliminary findings from SPICE phase 2 trials, of the two attributes at level 2 (managed), work product management (2.2) is more often rated lower than performance management (2.1); and at level 3 (established), process definition (3.1) is rated lower than the process resource attribute (3.2). (The achievement attributes are defined in Table 6.4.)

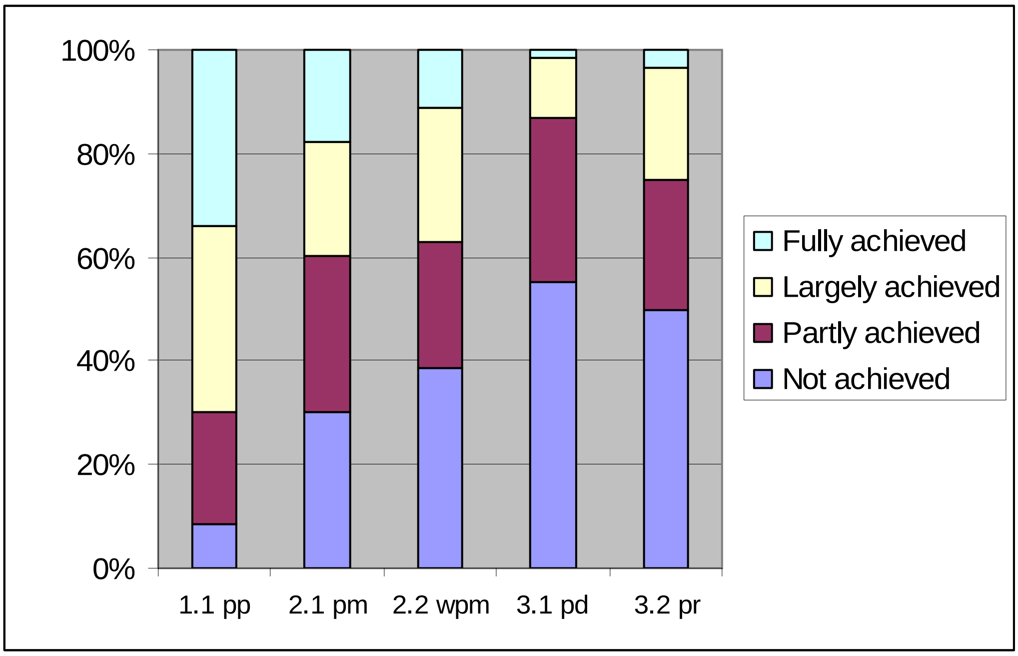


Figure 7-2 Attribute achievement ratings profile

Overall, there was a wide variation in the process capability levels for the 22 firms, as shown in Figure 7-3 (based on Table I.8 in Appendix I). The requirements elicitation process exhibited higher capability compared to the other processes in almost all cases. In all, 11 of the 22 firms were rated at level 2 (managed) or level 3 (established) for requirements elicitation. Software development and configuration management processes were also quite strong with a significant proportion of organisations assessed at either the managed or established level. On the other hand, the most incomplete process was process establishment, rated as level 0 (incomplete) at 15 of the 22 firms.

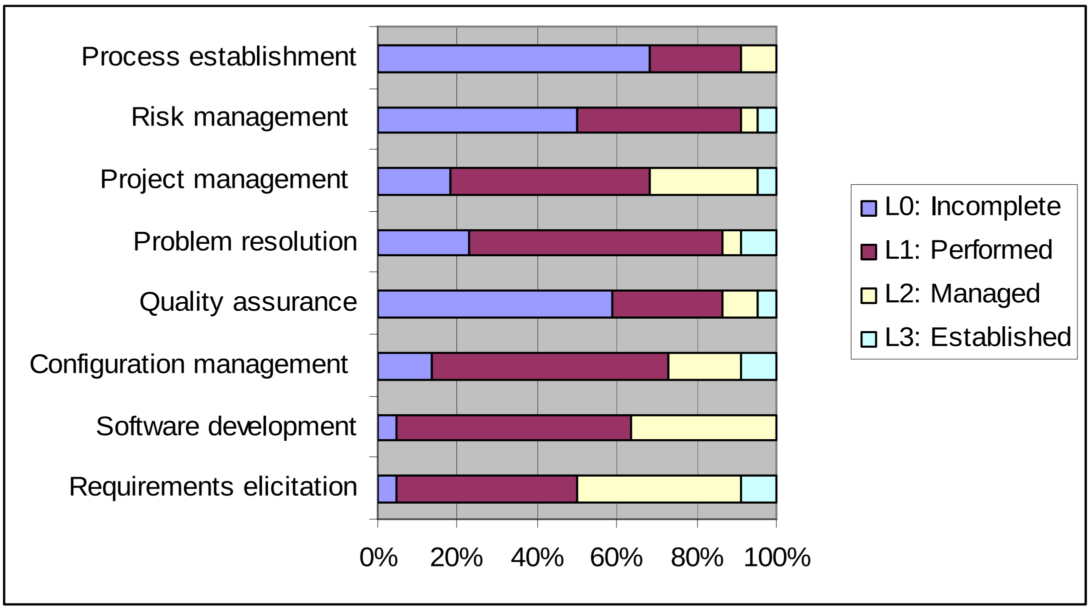


Figure 7-3 Capability levels by process

7.3.2 Variation in capability across processes

In order to determine the extent of variation in capability levels across the eight processes, Friedman's test (Selvanathan et al. 2000) was used. For each of the 22 firms, the eight processes were ranked from one to eight according to the capability level of the processes. The test statistic was calculated based on these ranks. The test confirmed that a significant difference existed across the eight process capability levels (χ^2=54.663; n=22; df=7; p<.05) (detailed test results in Appendix I Table I.9). The processes ranked in order of capability level are listed in Table 7.15.

Table 7.15 Processes ranked by capability level

Process		Mean of rank
RE	Requirements elicitation	6.27 [a]
SD	Software development	5.64 [a b]
CM	Configuration management	5.23 [a b c]
PM	Project management	5.11 [b c]
PR	Problem resolution	4.48 [c d]
RM	Risk management	3.48 [d e]
QA	Quality assurance	3.16 [d e]
PE	Process establishment	2.64 [e]
Note: rank means with the same superscript are not significantly different (p<.05).		

Pair-wise comparisons were made using Wilcoxon signed ranks tests (Selvanathan et al. 2000) to determine if a statistical difference existed between the capability levels of the eight processes. The analysis revealed five overlapping groups of processes (2-tailed tests, $p<0.05$, results are in Appendix I, Table I.10):

Group a: RE SD CM

Group b: SD CM PM

Group c: CM PM PR

Group d: PR RM QA

Group e: RM QA PE

For the attribute achievement measure, Friedman's test confirmed that a significant difference existed across the eight processes (χ^2=57.692; n=22; df=7; $p<0.05$) (detailed test results in Appendix I Table I.9). The processes ranked in order of attribute achievement are listed Table 7.16. The four highest ranking processes by attribute achievement occur in a different order compared to their ranking by capability level (in Table 7.15).

Table 7.16 Processes ranked by attribute achievement

Process		Mean of rank
SD	Software development	6.32 [a]
RE	Requirements elicitation	6.18 [a]
PM	Project management	5.11 [b]
CM	Configuration management	5.05 [b]
PR	Problem resolution	4.73 [b]
RM	Risk management	3.20 [c]
QA	Quality assurance	2.80 [c]
PE	Process establishment	2.61 [c]
Note: rank means with the same superscript are not significantly different (p<.05).		

Pair-wise comparisons using Wilcoxon signed ranks tests revealed three distinct groups of processes (2-tailed tests, $p<0.05$, results are in Appendix I, Table I.11):

Group a: SD RE

Group b: PM CM PR

Group c: RM QA PE

This calculation for attribute achievement supports the hypothesis that the attribute achievement of the primary lifecycle processes (software development and requirements elicitation) is statistically higher than the other groups of processes.

Therefore, the null hypothesis, that there is no difference in the eight process capability levels is rejected. It was hypothesised (in §2.7.2) that the primary lifecycle processes (software development and requirements elicitation) would be at higher capability levels than the support (configuration management, quality assurance and problem resolution) and organisation (project management, risk management and process establishment) processes. Although the capability levels of the requirements elicitation and software development processes are higher than the other processes, the difference between software development capability and configuration management capability level is not significant. *Therefore, there is limited support for hypothesis H2a, that the capability level of primary lifecycle processes is higher than the capability levels of support and organisation processes.*

For the attribute achievement measure, the analysis revealed three homogeneous groups of processes, but slightly different groupings to that found above for the capability levels. Processes within each group show little difference in their attribute achievement ranks. The group of processes exhibiting the highest attribute achievement consisted of software development and requirements elicitation. In the middle group, similar attribute achievement was found for project management, configuration management, and problem resolution processes. The group of processes with the lowest attribute achievement comprised quality assurance, risk management, and process establishment.

As can be seen from the groups presented in Table 7.15 and Table 7.16, the ranking and grouping of the achievement attribute measure is slightly different to that of the capability level measure: this difference will be discussed in the next chapter. *The attribute achievement grouping fully supports H2a: attribute achievement of the primary lifecycle processes (software development and requirements elicitation) is higher than the other processes.*

Note that in the following sections, where results of statistical tests are presented, significant outcomes are denoted in the following manner: (*) indicates the result is significant at p=.05; (**) at p=.01.

7.3.3 Relationship firm size and process capability

In §2.7.2, it was hypothesised that large firms are more likely than small firms to have higher process capability levels. As detailed in §7.2.1, 19 of the organisations had less than 20 full-time equivalent staff and can be classified as small. The other three firms are medium sized with a range from 55 to 65 staff FTE.

The interval variable staff FTE was compared with the eight process capability levels by use of the Spearman rank correlation coefficient. No relationship was found between the number of staff and the capability levels of any of the eight processes (Table 7.17). As the correlations are not significant at the .05 level, the null hypothesis is not rejected. *Hypothesis H2b—large firms are likely to have higher capability levels than small firms— is not supported.*

Process attribute achievement ratings were then compared against staff FTE and again no relationship was found (Table 7.17).

Table 7.17 Tests of association: process capability and firm size

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.168	.195	-.142	.289	-.003	.149	-.084	-.145
p (1-tailed)	.227	.193	.265	.096	.495	.255	.356	.260
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.180	-.025	.051	.356	-.017	.024	-.025	-.055
p (1-tailed)	.211	.456	.411	.052	.469	.458	.455	.404

7.3.4 Relationship proportion of experienced staff and process capability

It was predicted (in §2.7.2) that firms with a higher proportion of experienced staff would exhibit higher process capability levels. Computing Spearman rank correlation coefficients (r_s) for the ordinal variables 'proportion of experienced staff' and the eight process capability levels, a significant association was found for five of the eight processes: software development, configuration management, quality assurance, problem resolution, project management (results in Table 7.18). Therefore, the null hypothesis, that there is no difference in capability levels of firms with low proportions of experienced staff and capability levels of firms with high proportions of experienced staff, was rejected. *Thus, hypothesis H2c, that firms with experienced staff have higher capability levels, is supported for software development, configuration management, quality assurance, problem resolution, and project management.*

A similar comparison was undertaken to test the association of process attribute achievement with the proportion of experienced staff. With the same outcome as above, significant associations were found for five of the eight processes: software development, configuration management, quality assurance, problem resolution, project management (results in Table 7.18).

Table 7.18 Tests of association: process capability and proportion of experienced staff

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.118	.490(*)	.444(*)	.422(*)	.436(*)	.507(**)	.206	.198
p (1-tailed)	.305	.012	.022	.028	.024	.010	.186	.194
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.246	.623(**)	.523(**)	.451(*)	.494(*)	.388(*)	.308	.280
p (1-tailed)	.141	.001	.007	.020	.011	.041	.087	.110

7.3.5 Relationship extent of formal education of staff and capability level

It was hypothesised (in §2.7.2) that firms employing higher proportions of staff with formal education qualifications would exhibit higher capability levels compared to firms with lower proportions of educated staff. As described in §7.2.1, many firms reported large proportions of staff with graduate and post graduate qualifications. To investigate the relationship between education level and capability level, firstly, the proportion of staff with post graduate qualifications was compared with the process capability levels by calculating Spearman rank correlation coefficients. A significant association was found for requirements elicitation and quality assurance processes. Secondly, the proportion of staff with graduate or post graduate qualifications was compared with the process capability levels and no significant associations were found. Finally, the proportion of staff without formal qualifications was compared with overall capability level and again no significant associations were found. These tests are presented in Table 7.19.

Therefore, the null hypothesis, that there is no difference in capability levels of firms with low proportions of qualified staff and capability levels of firms with high proportions of qualified staff, was rejected. *Thus, hypothesis H2d, that firms with higher proportions of staff with post graduate qualifications have higher process capability levels, is supported for requirements elicitation and quality assurance processes.*

Table 7.19 Tests of association: capability levels and proportion of formally qualified staff

Post graduate	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.639(**)	.353	.111	.684(**)	.062	.337	.154	.332
p (1-tailed)	.001	.054	.312	.000	.391	.062	.246	.066
Graduate	RE	SD	CM	QA	PR	PM	RM	PE
r_s	-.110	.280	.351	.181	-.015	.277	-.072	.203
p (1-tailed)	.313	.104	.055	.210	.474	.106	.375	.183
No formal qualifications	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.110	-.280	-.351	-.181	.015	-.277	.072	-.203
p (1-tailed)	.313	.104	.055	.210	.474	.106	.375	.183

In a similar way, firms with a higher proportion of post graduate staff exhibited higher process attribute achievement for requirements elicitation, quality assurance and process establishment compared to those with less post graduate qualified staff. The detailed statistical tests are included in Table 7.20.

Table 7.20 Tests of association: attribute achievement and proportion of formally qualified staff

Post graduate	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.654(**)	.313	.248	.503(**)	.132	.292	.322	.427(*)
p (1-tailed)	.000	.078	.133	.009	.279	.093	.072	.024
Graduate	RE	SD	CM	QA	PR	PM	RM	PE
r_s	-.062	.334	.279	.146	-.011	.247	-.127	.221
p (1-tailed)	.393	.064	.104	.259	.481	.133	.286	.162
No formal qualifications	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.062	-.334	-.279	-.146	.011	-.247	.127	-.221
p (1-tailed)	.393	.064	.104	.259	.481	.133	.286	.162

7.3.6 Relationship proportion of technical staff and process capability

It was hypothesised (in §2.7.2) that firms employing higher proportions of technical staff would exhibit higher process capability levels compared to firms with lower proportions of technical staff. As described in §7.2.1, the proportion of technical staff to total staff at each firm was calculated. This variable, 'proportion of technical staff' was compared with the eight process capability levels by calculating a Spearman rank correlation coefficient. A significant association was found for four of the eight processes: software development, quality assurance, project management, and process establishment (statistical results in Table 7.21). Therefore, the null hypothesis, that there is no difference in capability levels of firms with low proportions of technical staff and capability levels of firms with high proportions of technical staff, was rejected. *Thus, hypothesis H2e, that firms with higher proportions of technical staff have higher capability levels, is supported for software development, quality assurance, project management, and process establishment.*

In a similar way, firms with a higher proportion of technical staff exhibited higher process attribute achievement compared to those with less technical staff for three of the eight processes: software development, quality assurance and project management (statistical results in Table 7.21).

Table 7.21 Tests of association: process capability and proportion of technical staff

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.205	.480(*)	.110	.458(*)	.134	.578(**)	.040	.398(*)
p (1-tailed)	.180	.012	.313	.016	.277	.002	.430	.033
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.312	.478(*)	.294	.506(**)	.193	.506(**)	.005	.275
p (1-tailed)	.079	.012	.092	.008	.195	.008	.491	.108

7.3.7 Relationship age of firm and process capability level

It was predicted (in §2.7.2) that more recently established firms would exhibit higher capability compared to firms which had been in operation for longer duration. The interval variable 'age of firm' was compared with the eight process capability levels by use of the Spearman rank correlation coefficient. No relationship was found between these variables (as shown in Table 7.22). As none of the correlations are significant at the .05 level, the null hypothesis is not able to be rejected. *Therefore, hypothesis H2f—process capability is associated with the length of time the firm has been in operation— is not supported.*

Process achievement was then compared against the age of firm and again no relationship was found (statistical results in Table 7.22).

Table 7.22 Tests of association: process capability and age of firm

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.141	-.057	-.167	.068	-.371	-.018	-.010	.063
p (1-tailed)	.288	.411	.254	.395	.065	.472	.485	.402
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	-.057	-.154	-.031	.170	-.182	-.083	.024	.032
p (1-tailed)	.411	.271	.452	.250	.235	.372	.462	.449

7.3.8 Relationship target business sector and process capability

In §2.7.2, it was predicted that process capability may vary depending on the business sector targeted by the firm. As reported in §7.2.1, many of the 22 firms selected more than one target business sector. To compare the capability levels of firms according to their target business sector, the Kruskal-Wallis test was used. Firstly, the sectors were ranked for each process according to their capability level and also according to their attribute achievement. The results, summarised in Table 7.23, indicate that overall, the sectors ranked most highly are public utilities and administration, telecommunications and media, and defence and aerospace.

Table 7.23 Top ranking target business sector by process capability for each process

Process	Top ranking target business sector	
	By capability level	By attribute achievement
Requirements elicitation	Defence and aerospace	Defence and aerospace
Software development	Public utilities and administration, telecommunications and media	Public utilities and administration
Configuration management	Public utilities and administration	Public utilities and administration
Quality assurance	Public utilities and administration	Defence and aerospace
Problem resolution	Construction, mining, petroleum, agriculture	Construction, mining, petroleum, agriculture
Project management	Telecommunications and media	Telecommunications and media
Risk management	Education, health and pharmaceutical	Telecommunications and media
Process establishment	Telecommunications and media	Telecommunications and media

However, the Kruskal-Wallis test indicates that no significant difference exists among the various sector groups for either capability level or attribute achievement (rankings in Appendix I Table I.12, test result in Table 7.24).

Table 7.24 Comparison of process capability by target business sectors

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
χ^2	8.600	8.547	7.541	11.641	5.524	6.628	6.408	11.329
df	9	9	9	9	9	9	9	9
p	.475	.480	.581	.234	.786	.676	.698	.254
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
χ^2	4.872	9.859	8.246	7.394	6.835	6.551	3.966	7.894
df	9	9	9	9	9	9	9	9
p	.845	.362	.510	.596	.654	.684	.914	.545

To further investigate the relationship between target business sectors and process capability, the sectors were separated into two groups: firms with publicly funded clients, and firms without clients from the public sector. Public administration, public utilities, defence and telecommunications target business sectors were coded as publicly funded (value 1), and the other sectors as privately funded (value 2). The Mann-Whitney U test was used to determine whether the two independent groups came from the same population. Based on the process capability levels, no significant differences were found between the public sector group and the non-public sector group (rankings in Appendix I Table I.13, test results in Table 7.25).

For attribute achievement, a significant difference was reported: the risk management process attribute achievement for firms targeting public sector clients ranked higher than firms not targeting public sector clients (rankings in Appendix I Table I.14, test results in Table 7.25). *Therefore, hypothesis H2g—process capability is associated with specific target business sectors, is supported for risk management process and firms targeting public sector clients.*

Table 7.25 Comparison of process capability for firms with public and non-public sector clients

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
Mann-Whitney U	35.000	31.500	48.000	39.000	44.500	37.500	32.000	51.500
Wilcoxon W	71.000	67.500	84.000	75.000	80.500	73.500	68.000	87.500
Z	-1.565	-1.935	-.616	-1.319	-.918	-1.369	-1.822	-.375
p (2-tailed)	.118	.053	.538	.187	.359	.171	.068	.708
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
Mann-Whitney U	27.500	49.000	37.500	34.500	39.500	44.500	24.500	46.500
Wilcoxon W	63.500	85.000	73.500	70.500	75.500	80.500	60.500	82.500
Z	-1.955	-.482	-1.270	-1.492	-1.132	-.789	-2.167	-.661
p (2-tailed)	.051	.630	.204	.136	.258	.430	.030	.508

It was also considered than firms focussing on only one or two business sectors may exhibit higher process capability compared to firms servicing a broad range of sectors. The 11 firms which target one or two sectors were coded separately from those servicing three or more sectors. The Mann-Whitney U test found no significant associations for capability levels or attribute achievement (rankings in Appendix I Table I.15 and Table I.16, test results in Table 7.26).

Table 7.26 Comparison of process capability of firms with few or many target business sectors

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
M-W U	49.000	53.500	45.000	58.500	60.000	59.000	60.000	42.000
Wilcoxon W	115.000	119.500	111.000	124.500	126.000	125.000	126.000	108.000
Z	-.825	-.532	-1.148	-.149	-.038	-.107	-.037	-1.482
p (2-tailed)	.410	.595	.251	.881	.969	.915	.971	.138
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
M-W U	57.500	55.000	46.000	56.500	48.500	60.000	53.500	55.500
Wilcoxon W	123.500	121.000	112.000	122.500	114.500	126.000	119.500	121.500
Z	-.198	-.365	-.958	-.267	-.792	-.033	-.463	-.335
p (2-tailed)	.843	.715	.338	.789	.428	.974	.643	.738

7.3.9 Relationship ISO 9001 certification and process capability

In §2.7.2, it was predicted that firms with ISO 9001 certification would have higher process capability levels compared to firms without certification. However, caution should be exercised in interpreting the result of the statistical test as only 2 of the 22 firms were certified to ISO 9001 (§7.2.1). Data were coded such that firms without certification were given a value of zero, and firms with certification coded as one.

A series of Mann-Whitney U tests revealed a significant association between certification to ISO 9001 and the capability levels of six of the eight processes: requirements elicitation, software development, project management, risk management, quality assurance and process establishment (statistical tests in Table 7.27). *Thus, hypothesis H2h, that firms with ISO 9001 have higher capability levels, is supported for six of the eight processes: requirements elicitation, software development, project management, risk management, quality assurance and process establishment.*

In a similar way, firms with ISO 9001 certification exhibited higher process achievement compared to those without certification for requirements elicitation, quality assurance, risk management and process establishment (Table 7.27).

Table 7.27 Comparison of process capability based on ISO 9001 certification

Capability Level	RE	SD	CM	QA	PR	PM	RM	PE
M-W U	5.500	6.000	14.500	4.500	18.000	6.000	5.000	7.000
Wilcoxon W	215.500	216.000	224.500	214.500	228.000	216.000	215.000	217.000
Z	-1.808	-1.850	-.709	-2.012	-.267	-1.734	-1.906	-1.811
p (1 tailed)	.036(*)	.032(*)	.240	.022(*)	.395	.042(*)	.029(*)	.035(*)
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
M-W U	4.000	7.000	9.000	1.500	12.000	7.000	2.000	1.500
Wilcoxon W	214.000	217.000	219.000	211.500	222.000	217.000	212.000	211.500
Z	-1.837	-1.499	-1.263	-2.148	-.919	-1.493	-2.072	-2.155
p (1-tailed)	.033(*)	.067	.103	.016(*)	.179	.068	.019(*)	.016(*)

7.3.10 Relationship project characteristics and process capability

In §2.7.2, hypotheses about the association between process capability and various projects characteristics were formulated. In this section, the statistical tests related to these project variables are reported.

Number of projects in progress. It was expected that a positive association may exist between the number of projects in progress at each firm and process capability, but the reverse may be the case. A negative correlation of capability level with number of projects in progress for the configuration management process indicated that firms with many projects in progress exhibited lower capability for configuration management than firms with few current projects (Table 7.28). *Therefore, the null hypothesis for H2i is rejected, as there is support for the hypothesis that a negative association exists between the capability of the configuration management process and the number of projects in progress.*

The number of projects in progress was then compared against the eight process attribute achievement ratings but no relationship was found (Table 7.28).

Table 7.28 Tests of association: process capability and number of projects in progress

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.029	-.153	-.537(*)	-.064	-.192	.082	.088	.158
p (1-tailed)	.901	.507	.012	.783	.406	.723	.706	.495
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	-.050	-.102	-.293	-.024	-.242	.107	-.068	-.008
p (1-tailed)	.830	.661	.197	.917	.290	.645	.769	.974

Typical number of employees per project. It was expected (§2.7.2) that firms with projects with large numbers of staff would exhibit higher process capability levels, but no relationship was found (Table 7.29). Therefore, the null hypothesis is not rejected. *Hypothesis H2j—firms with large project teams are likely to have higher capability levels than firms with projects staffed by a small number of staff—is not supported.*

However, when the number of staff per project was compared with the eight process attribute achievements, a significant relationship was found for number of employees per project and the achievement attribute of the project management process (Table 7.29).

Table 7.29 Tests of association: process capability and number of employees per project

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.096	.315	.129	.254	.245	.238	.175	.014
p (1-tailed)	.339	.082	.289	.133	.142	.149	.224	.476
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.230	.259	.192	.318	.280	.375(*)	.278	.284
p (1-tailed)	.158	.128	.203	.080	.110	.047	.111	.106

Typical duration to final delivery. In §2.7.2, it was predicted that projects with longer durations would exhibit higher process capability levels. Computing Spearman rank correlation coefficients for the interval variable ‘project duration’ and the eight process capability levels revealed significant associations for software development, quality assurance, problem resolution and project management (Table 7.30). Therefore, the null hypothesis, that there is no difference in the capability levels of firms with long duration projects and firms with short duration projects, was rejected. *Thus, hypothesis H2k—firms with long duration projects are likely to have higher capability levels than firms with short duration projects—is supported for software development, quality assurance, problem resolution and project management.*

Process attribute achievement was then compared against the duration of projects and significant associations were found for all processes with the exception of risk management (Table 7.30).

Table 7.30 Tests of association: process capability and project duration

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.365	.430(*)	.347	.568(**)	.397(*)	.386(*)	-.095	.245
p (1-tailed)	.052	.026	.062	.004	.037	.042	.341	.142
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.492(*)	.589(**)	.472(*)	.431(*)	.500(*)	.398(*)	.305	.538(**)
p (1-tailed)	.012	.002	.015	.025	.011	.037	.089	.006

Average cost overrun. In responding to the demographic questionnaire, only 13 of the 22 firms answered the question about the typical project cost overrun. In §2.7.2, it was expected that firms with projects excessively over budget would exhibit lower process capability levels, but no relationship was found (Table 7.31). Therefore, the null hypothesis is not rejected. *Hypothesis H2l—firms with projects with high cost overruns are likely to have lower capability levels than firms with projects completed on budget—is not supported.*

The process attribute achievement measures were then compared against project cost overrun and again no relationship was found (Table 7.31).

Table 7.31 Tests of association: process capability and average cost overrun

Capability level	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.104	-.102	.443	-.320	-.106	-.397	-.020	-.311
p (1-tailed)	.367	.371	.065	.143	.365	.089	.475	.151
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.052	-.327	.181	-.283	-.014	-.316	-.048	-.032
p (1-tailed)	.433	.138	.277	.175	.482	.147	.439	.459

7.3.11 Relationship perceptions of performance and process capability

Earlier, in §7.2.4, summaries were provided of responses to the two questions about the sponsors' perceptions of performance and the importance of performance measures. In §2.7.2, it was predicted that process capability would be positively associated with perceived performance.

As stated in §6.9.4, the following scale was used to recode the data: excellent—4; good—3; fair—2; poor—1; and 'don't knows' were coded as missing values. The process capability levels were compared with the perceptions of the sponsors about their firm's performance in terms of ability to meet budget and schedule commitments, customer satisfaction and meeting requirements, staff productivity and morale. Spearman rank correlations were calculated to determine the association between each of the six performance variables and the eight process capability levels.

Table 7.32 Tests of association: capability levels and performance perceptions

Measure		RE	SD	CM	QA	PR	PM	RM	PE
Budget	r_s	0.046	0.175	-0.31	0.373	0.074	.522(*)	0.004	0.373
n=19	*p*	0.427	0.237	0.098	0.058	0.381	0.011	0.494	0.058
Schedule	r_s	0.145	0.274	-0.114	0.317	0.143	.554(**)	0.265	.380(*)
n=22	*p*	0.26	0.108	0.307	0.075	0.263	0.004	0.117	0.04
Customer	r_s	0.232	0.36	-0.159	.391(*)	0.336	0.262	-0.316	0.177
n=20	*p*	0.162	0.06	0.251	0.044	0.074	0.132	0.087	0.228
Requirements	r_s	-0.075	-0.306	-0.171	0.078	-0.011	-0.257	-0.342	-0.116
n=22	*p*	0.369	0.083	0.223	0.364	0.481	0.124	0.06	0.304
Productivity	r_s	0.092	0.071	-0.187	0.105	0.088	0.129	-0.349	-0.142
n=22	*p*	0.342	0.376	0.203	0.32	0.348	0.284	0.056	0.265
Morale	r_s	0.259	0.131	-0.337	0.07	-0.143	0.3	0	0.26
n=21	*p*	0.129	0.286	0.068	0.381	0.268	0.093	0.5	0.127

Note: all tests 1-tailed.

The results, summarised in Table 7.32, indicate that budget performance is associated with project management capability levels, schedule performance is associated with project management and process establishment, and customer performance is associated with quality assurance. *Therefore, for hypothesis H2m, some of the perceived performance measures are positively associated with process capability levels for quality assurance, project management and process establishment processes.*

In a similar way, the eight process attribute achievements were compared with the six perceived performance measures. As shown in Table 7.33, schedule performance is associated with quality assurance, project management and process establishment; customer performance with software development; and requirements performance with project management.

Table 7.33 Tests of association: attribute achievement and performance perceptions

Measure		RE	SD	CM	QA	PR	PM	RM	PE
Budget	r_s	.094	.357	-.095	.242	-.023	.388	.043	.247
	p	.350	.067	.349	.159	.463	.050	.431	.154
Schedule	r_s	.172	.293	.095	.437(*)	.023	.618(**)	.294	.373(*)
	p	.222	.093	.337	.021	.460	.001	.092	.044
Customer	r_s	.231	.441(*)	.112	.290	.287	.111	-.024	.089
	p	.164	.026	.320	.107	.110	.320	.460	.354
Requirements	r_s	-.226	-.139	-.150	-.020	-.007	-.411(*)	-.117	-.174
	p	.156	.269	.253	.464	.487	.029	.302	.219
Productivity	r_s	.196	.202	-.186	.051	-.118	.058	-.290	-.273
	p	.191	.183	.204	.410	.300	.399	.096	.109
Morale	r_s	.350	.265	-.235	-.098	-.132	.332	.075	.208
	p	.060	.123	.153	.336	.284	.071	.374	.182

Note: all tests 1-tailed.

The significant associations between capability levels and performance perceptions and also attribute achievement and performance perceptions are summarised in Table 7.34.

Table 7.34 Summary of associations: process capability and performance perceptions

Perceptions relating to firm's …	N	RE	SD	CM	QA	PR	PM	RM	PE
Ability to meet budget commitments	19						X		
Ability to meet schedule commitments	22				Y		XY		XY
Ability to achieve high customer satisfaction	20		Y		X				
Ability to meet specified and implied requirements	22						Y		
Staff productivity	22								
Staff morale/job satisfaction	21								

X indicates that a significant relationship exists between capability level and performance indicators
Y indicates that a significant relationship exists between attribute achievement and performance indicators

Sponsors were also asked to rate the importance of six performance measures. The following scale was used to recode the data: very important—4; important—3; somewhat important—2; not important—1. The eight process capability levels were compared against these perceptions of the importance of the performance measures. The detailed results, provided in Appendix I Table I.17 and summarised in Table 7.35, indicate three associations: between the importance of budget performance and risk management; and customer satisfaction with both quality assurance and problem resolution.

A similar method was applied to compare the process attribute achievement with the perceptions of the importance of performance measures and two associations were found: meeting requirements with risk management, and staff morale with software development (summarised in resulted Table 7.35, results in Appendix I Table I.17).

Table 7.35 Summary of associations: process capability and perceptions about the importance of performance indicators

Performance indicator	N	RE	SD	CM	QA	PR	PM	RM	PE
Importance of budget	22							X	
Importance of schedule	22								
Importance of customer	22				X	X			
Importance of requirements	22							Y	
Importance of productivity	22								
Importance of morale	22		Y						

X indicates that a significant relationship exists between capability level and importance of performance indicators
Y indicates that a significant relationship exists between attribute achievement and importance of performance indicators

7.3.12 Summary of capability assessments

A summary of the hypothesis tests associated with research question 2 is shown in Table 7.36. In Chapter 8, the findings will be discussed and conclusions drawn.

Table 7.36 Summary of significant correlations of organisation and project characteristics with process capability

Organisation/project characteristic	N	RE	SD	CM	QA	PR	PM	RM	PE
Size – headcount	18								
Proportion of experienced staff	21		XY	XY	XY	XY	XY		
Proportion of post graduate staff	22	XY			XY				Y
Proportion of technical staff	22		XY		XY		XY		X
Age of firm	18								
Target business sector – public sector clients	22							Y	
ISO 9001	22	XY	X		XY		X	XY	XY
Projects in progress	21			X					
Number of staff/project	21						Y		
Project duration	21	Y	XY	Y	XY	XY	XY		Y
Cost overrun	13								

X indicates that a significant relationship exists for capability level
Y indicates that a significant relationship exists for attribute achievement

In the next section, the results of the hypotheses related to research question 3 are reported.

7.4 Process capability at follow-up meetings

The third and final research question asked: are assessment-based SPI programs an effective means to improve process capability for small software development firms, and is the extent of improvement associated with organisational characteristics? The procedures relating to the follow-up meetings and compilation of the final reports are described in §6.6. The final reports were analysed and the process capability at the end of the process improvement program compared with that determined at the initial assessments. Only two organisations withdrew from the program: one firm cancelled its SEA membership after the initial assessment (firm #24); the other firm could not be contacted as it had ceased to operate prior to the follow-up meeting (#2). These two firms were excluded from the analysis of the extent of improvement as it was not known if any of their processes had improved.

7.4.1 Capability level improvement

Analysis of the 20 reports from the follow-up meetings revealed that six firms had improved the capability levels of their processes. A summary of capability levels as determined at the follow-up meetings is presented in Table 7.37. A summary of each of the assessment and final reports is included in Appendix H, and the list of capability levels and attribute achievement ratings at the follow-up meeting is in Appendix I Table I.18.

Table 7.37 Frequency of capability levels by process at follow-up meetings

Process	Process capability level			
	Level 0 Incomplete	Level 1 Defined	Level 2 Managed	Level 3 Established
Requirements elicitation	1	8	8	3
Software development	1	12	4	3
Configuration management	2	10	6	2
Quality assurance	11	5	3	1
Problem resolution	4	12	1	3
Project management	3	10	4	3
Risk management	9	9	1	1
Process establishment	11	5	4	0
Total at follow-up meetings - 20 firms	42	71	31	16
(Total at initial assessment 20 firms)	(48)	(73)	(30)	(9)

A comparison of the capability levels at the initial assessments (Appendix I Table I.8) and the follow-up meetings (Table 7.37) is provided in Figure 7-4.

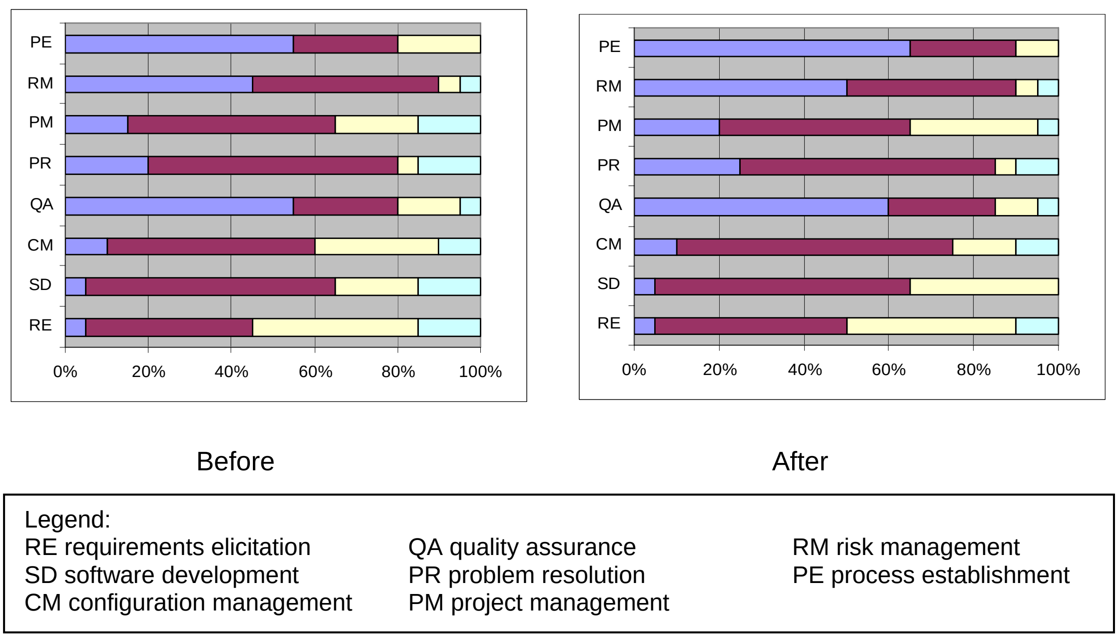


Figure 7-4 Comparison of capability levels at assessment and follow-up meetings

It was planned to hold each follow-up meeting six months after the initial assessment to determine the extent to which each firm had implemented the recommendations. The actual elapsed time between the initial assessment and follow-up meeting ranged from seven to 16 months.

Six of the firms recorded changes to process capability levels as shown in Table 7.38, listing the firms in decreasing order of improvement. Two firms reported an outstanding result: firm #9 improved a total of nine levels across six processes, and firm #8 improved a total of eight levels across seven processes.

Table 7.38 Capability levels for six most improved firms at follow-up meeting

Firm	RE	SD	CM	QA	PR	PM	RM	PE	Total processes	Total levels
#9	1→2	2→3	3	0→2	3	2→3	0→2	0→2	6	9
#8	2→3	2→3	1→2	1→2	1→3	1→2	3	1→2	7	8
#18	3	2→3	1→2	2	1	2	0→1	1→2	4	4
#5	1	1	1	0→1	1	0→1	0	0	2	2
#13	1	0	1	0	0→1	1	0	0→1	2	2
#21	1	1	1→2	0	1	1	1	0	1	1
Total Firms	2	3	3	3	2	3	2	4	22	
Total Levels	2	3	3	4	3	3	3	5		26
Note: arrow denotes improvement in capability level from initial assessment to follow-up meeting										

Examining the extent of improvement across all eight processes, capability levels improved in all processes, with the process exhibiting the lowest capability at the time of the assessments, *process establishment*, improving more than the other processes. As shown in Table 7.38, the process establishment process increased a total of five levels, with one firm (#13) increasing from level 0 (incomplete) to level 1 (performed); one firm (#9) from level 0 to level 2 (managed); and one firm (#18) from level 1 to level 2. Quality assurance improved two levels from level 0 to 2 at firm #9, and one level at firm #8 and #5. Three firms (#9, #8, #18) improved their software development process from level 2 to level 3 (established). Configuration management also improved in three firms, all from level 1 to level 2 (firms #8, #18, #21). Project management improvement was more varied: firm #5 from level 0 to 1; firm #8 from level 1 to 2; and firm #9 from level 2 to 3. Risk management improved two levels by firm #9 from 0 to 2, and firm #18 from level 0 to 1. The process with the highest capability at the initial assessments showed the least improvement: firm #9 improved requirements elicitation process from level 1 to 2; and firm #8 from level 2 to 3.

The capability levels of 22 processes changed as shown in Table 7.38, five processes improved from 0 to 1; eight processes from level 1 to 2, and five processes from level 2 to 3. Also, there were four instances where process capability level increased two levels: three processes improved from level 0 to 2; and one process from 1 to 3. Therefore, overall 22 process instances improved providing a total improvement of 26 levels.

As summarised in Table 7.39, nine firms were formally reassessed, and six of these had improved their process capability levels, the other three exhibited improvements, but not enough to gain a higher capability level rating. A further 11 firms participated in the follow-up meetings, but were not formally reassessed. Of this group (informally reassessed), six firms reported that they had implemented some of the recommendations; and five firms did not report any improvement, but provided reasons why the recommendations had not been actioned. The experiences of these firms is discussed in more detail in §8.4.1.

Table 7.39 Extent of improvement by firms

Group	Number of firms	Follow-up meeting	Firm ID#	Extent of improvement
1	6	Formal reassessment	5,8,9,13,18,21	Capability levels
2	1	Formal reassessment	23	Attribute achievement
3	2	Formal reassessment	14,16	Improvement to specific processes
4	6	Informal reassessment	1,2,7,17,19,25	Limited improvement
5	5	Informal reassessment	4,11,12,15,22	No improvement reported
6	2	No follow-up meeting	2, 24	Unknown

7.4.2 Achievement attribute improvement

The following Table 7.40 shows the seven firms which make up groups 1 and 2; these firms all improved their attribute achievement. In two of the firms, the attribute achievement improved, but did not alter the capability level. These two instances are highlighted with an asterisk in Table 7.40 Firm #9 improved achievement of process definition (PA3.1) attribute for configuration management from largely to fully (this did not change the process capability level, it was already at level 3). In firm #23, the performance management attribute (PA2.1) of the project management process improved from partially to largely achieved.

Table 7.40 Attribute achievement improvement

Firm	RE	SD	CM	QA	PR	PM	RM	PE	Total Attribute achievement points	Processes
#9	1	2	*1	8	0	1	8	6	27	7
#8	2	3	1	1	1	1	0	1	10	7
#18	0	1	2	0	0	0	1	3	7	4
#5	0	0	0	2	0	1	0	0	3	2
#21	0	0	3	0	0	0	0	0	3	1
#13	0	0	0	0	1	0	0	1	2	2
#23	0	0	0	0	0	*1	0	0	1	1
Note: * denotes improvement in attribute achievement rating, but no corresponding change in capability level.										

In Table 7.39, group 3 consists of two firms: the final report for firm #14 included comments about improving the process performance attribute (PA1.1) of quality assurance and process establishment (both were initially assessed at partially); and firm #16's final report included positive comments related to process performance attribute (PA1.1) for quality assurance and configuration management (both initially assessed at partially), but none of these improvements were deemed by the assessor to be sufficient to change the process attribute achievement from partially to largely achieved.

7.4.3 Comparison of capability at assessment and follow-up meetings

In §2.7.3, it was predicted that the process capability levels at the time of the follow-up meetings would be higher than the process capability levels recorded at the initial assessment. To test the hypothesis, the non-parametric pre-test/post-test using the Wilcoxon signed rank test was used (Bonate 2000). This test found a significant difference in the process capability levels from the time of the initial assessment to the follow-up meetings for four of the eight processes: software development, configuration management, project management, and process establishment (results in Table 7.41).

Table 7.41 Comparison of capability levels and attribute achievement at assessment and follow-up meeting

	Difference from initial assessment to follow-up meeting							
Capability Levels	RE	SD	CM	QA	PR	PM	RM	PE
Z	1.414	1.732(*)	1.732(*)	1.633	1.342	1.732(*)	1.342	1.890(*)
p (1-tailed)	.078	.041	.041	.051	.090	.041	.090	.029
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
Z	1.342	1.604	1.841(*)	1.604	1.414	2.000(*)	1.342	1.841(*)
p (1-tailed)	.090	.054	.033	.055	.078	.023	.090	.033

Therefore, the null hypothesis, that there is no difference in the process capability levels at the time of the follow-up meetings, is rejected. *Thus, hypothesis H3a, that the process capability levels at the time of the follow-up meetings is greater than the process capability levels at assessment, is supported for software development, configuration management, project management, and process establishment processes.*

A similar comparison was undertaken using the process attribute achievement ratings and a significant difference was found for configuration management, project management and process establishment (based on 1 tailed tests) (results in Table 7.41).

7.4.4 Effect of firm size on extent of improvement

It was predicted in §2.7.3 that firms with larger staff numbers would benefit more than smaller firms from participating in the RAPID program. The extent of improvement of the eight process capability levels was compared to the staff FTE by calculating the Spearman rank-correlation coefficient. As shown in Table 7.42, no relationship was found.

Table 7.42 Tests of association: extent of process improvement and firm size

Difference in capability levels	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.043	.097	.353	-.079	.324	-.061	-.113	.128
p (1-tailed); n=20	.428	.342	.064	.370	.082	.400	.317	.295
Difference in attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.063	.109	.197	-.093	.318	.130	-.113	.121
p (1-tailed); n=20	.396	.324	.202	.348	.086	.292	.317	.305

As the association was not significant at the .05 level, the null hypothesis was not rejected. *Hypothesis H3b—the extent of improvement in process capability is associated with firm size—is not supported.*

The increase in process attribute achievement was then compared against staff FTE and again no relationship was found (details in Table 7.42).

7.4.5 Experience of staff and extent of improvement

In §2.7.3, it was hypothesised that firms with higher proportions of experienced staff would show greater improvement in overall process capability levels compared to firms with lower proportions of experienced staff. Computing a Spearman rank correlation coefficient for the ordinal variables 'proportion of experienced staff' and the extent of improvement in the eight process capability levels, no association was found (results in Table 7.43).

Table 7.43 Tests of association: extent of process improvement and proportion of experienced staff

Difference in capability levels	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.126	.159	.053	.175	.073	.185	.059	.089
p (1-tailed) (n=19)	.304	.258	.415	.237	.384	.224	.405	.358
Difference in attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.132	.163	.025	.173	.063	.095	.059	.091
p (1-tailed) (n=19)	.295	.252	.460	.239	.399	.350	.405	.355

As the association is not significant at the .05 level, the null hypothesis was not rejected. *Hypothesis H3c—the extent of improvement in process capability is associated with the proportion of experienced staff—is not supported.*

The increase in process attribute achievement was then compared against the proportion of staff with greater than five years experience and again, no relationship was found (details in Table 7.43).

7.4.6 Formal education of staff and extent of improvement

In §2.7.3, it was hypothesised that firms with higher proportions of formally qualified staff would show greater improvement in overall process capability levels compared to firms with lower proportions of qualified staff. Computing Spearman rank correlation coefficients for the ordinal variables proportion of staff with graduate or post graduate qualifications and the extent of improvement in the eight process capability levels, an association was found for the proportion of post graduate staff and the extent of improvement in the configuration management process, and also the proportion of graduate staff (with graduate or post graduate qualifications) and the extent of improvement in the process establishment process (results in Table 7.44).

Table 7.44 Correlations for extent of improvement with formal qualifications

Capability level		RE	SD	CM	QA	PR	PM	RM	PE
Post graduate	r_s	.091	.267	.420(*)	-.081	.112	-.064	.041	.083
	p (1-tailed); n=20	.352	.127	.033	.367	.319	.395	.432	.365
Graduate	r_s	.272	.342	.011	.157	.272	.148	.272	.404(*)
	p (1-tailed); n=20	.110	.060	.480	.243	.110	.255	.110	.031

Thus, hypothesis H3d, that the extent of improvement in process capability is associated with the proportion of post graduate staff, is supported for the configuration management process; and the extent of improvement in process capability is associated with the proportion of graduate staff, is supported for the process establishment process.

The increase in process attribute achievement was then compared against the proportion of staff with graduate or post graduate qualifications and a significant associations was found for the proportion of graduate staff and the extent of improvement in the attribute achievement for the process establishment process (details in Table 7.45).

Table 7.45 Correlations for extent of improvement with formal education qualifications

Attribute achievement		RE	SD	CM	QA	PR	PM	RM	PE
Post graduate	r_s	.112	.269	.252	-.099	.091	.023	.041	.096
	p (1-tailed); n=20	.319	.126	.142	.339	.352	.462	.432	.344
Graduate	r_s	.272	.341	.071	.148	.272	.081	.272	.404(*)
	p (1-tailed);n=20	.110	.060	.377	.256	.110	.360	.110	.031

7.4.7 Proportion of technical staff and extent of improvement

In §2.7.3, it was hypothesised that firms with higher proportions of staff in a technical role (rather than support or administrative function) would show greater improvement in overall process capability levels compared to firms with lower proportions of technical staff. Computing Spearman rank correlation coefficients for the ordinal variables 'proportion of technical staff' and the extent of improvement in the eight process capability levels, an association was found for the proportion of technical staff and the extent of improvement in the software development and process establishment processes.

Table 7.46 Tests of association: extent of process improvement and proportion of technical staff

Difference in capability levels	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.247	.415(*)	.219	.065	.349	.061	.295	.472(*)
p (1-tailed) (n=20)	.147	.035	.176	.393	.066	.399	.103	.018
Difference in attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	.253	.408(*)	.209	.050	.348	.261	.295	.476(*)
p (1-tailed) (n=20)	.141	.037	.188	.417	.066	.133	.103	.017

Thus, hypothesis H3e, that the extent of improvement in process capability is associated with the proportion of technical staff, is supported for the software development and process establishment processes.

The increase in process attribute achievement was then compared against the proportion of technical staff and again, an association was found for the proportion of technical staff and the extent of improvement in the software development and process establishment processes (details in Table 7.46).

Therefore, as shown in Table 7.47, although firms improved in capability and attribute achievement in processes such as software development, configuration management, quality assurance, project management and process establishment, the only organisation factor associated with the extent of improvement was the proportion of formally educated staff.

Table 7.47 Summary of significant correlations related to extent of improvement

Factor	N	RE	SD	CM	QA	PR	PM	RM	PE
Before/After	20		X	XY			XY		XY
Firm Size	20								
Proportion of Experienced Staff	19								
Proportion of post graduate staff Proportion of graduate staff	20			X					 XY
Proportion of Technical Staff	20		XY						XY
X indicates that a significant relationship exists between capability level and factor Y indicates that a significant relationship exists between attribute achievement and factor									

7.5 Summary of hypotheses and statistical tests

As shown in the summary of the results of the hypotheses (Table 7.48 and Table 7.49), support was found for many of the hypotheses relating to research questions 2 and 3. In §6.9.1, two measures of process capability were defined: capability level and attribute achievement. As highlighted with shading in Table 7.48 and Table 7.49, these two measures sometimes provided different results in the hypothesis tests. In five hypothesis tests (H2a, H2d, H2j, H2k, H2m), the attribute achievement measure proved to be slightly more discriminating than the capability level measure. This was expected as attribute achievement has a finer granularity and greater range (0-15 compared to capability level range 0-3) and as it recognises achievement beyond the capability level rating. However, for five other hypotheses (H2e, H2h, H2i, H3a, H3d), the process capability level was found to be significantly associated with organisation or project characteristics while the attribute achievement was not. The differences in the hypothesis outcomes, as listed in Table 7.48 and Table 7.49, may be more a result of the small sample size rather than strong associations, for example, only two firms were certified to ISO 9001 (H2h).

Table 7.48 Summary of statistical tests for research question 2

RQ2 Is the capability of technical processes higher than that of management processes, and is process capability is associated with particular organisational characteristics?		
Hypothesis in alternate form	Capability level	Attribute achievement
H2a the capability of primary lifecycle processes is higher that the capability of support and organisation processes	Limited support	Fully supported
H2b Process capability is positively associated with the size of the firm	Not supported	Not supported
H2c Process capability is positively associated with the proportion of experienced staff	Supported for SD, CM, QA, PR, PM	Supported for SD, CM, QA, PR, PM
H2d Process capability is positively associated with the proportion of post graduate qualified staff	Supported for RE, QA	Supported for RE, QA, PE
H2e Process capability is positively associated with the proportion of technical staff	Supported for SD, QA, PM, PE	Supported for SD, QA, PM
H2f Process capability is positively associated with the length of time the firm has been in operation	Not supported	Not supported
H2g Process capability is positively associated with specific industry sectors	Not supported	Not supported
H2h Process capability is positively associated with certification to ISO 9001	Supported for RE, SD, QA, PM, RM, PE	Supported for RE, QA, RM, PE
H2i Process capability is *negatively* associated with number of projects in progress	Supported for CM	Not supported
H2j Process capability is positively associated with size of the project team	Not supported	Supported for PM
H2k Process capability is positively associated with the typical time duration of the project.	Supported for SD, QA, PR, PM	Supported for RE, SD, CM, QA, PR, PM, PE
H2l Process capability is positively associated with the typical cost overrun of projects.	Not supported	Not supported
H2m Process capability is positively associated with perceptions of higher firm performance.	Supported: budget with PM; schedule with PM, PE; customer with QA	Supported: customer with SD; schedule with QA, PM, PE; customer with SD; requirements with PM

Table 7.49 Summary of statistical tests for research question 3

RQ3 Are assessment-based SPI programs an effective means to improve process capability for small software development firms, and is the extent of improvement associated with particular organisational characteristics?		
Hypothesis in alternate form	Capability level	Attribute achievement
H3a Process capability after the assessment (at the follow-up meeting) is higher compared to the initial assessment	Supported for SD; CM; PM; PE	Supported for CM; PM; PE
H3b The extent of improvement will be positively associated with the size of the firm	Not supported	Not supported
H3c The extent of improvement will be positively associated with the proportion of experienced staff	Not supported	Not supported
H3d The extent of improvement will be positively associated with the proportion of formally qualified staff	Supported for proportion of post graduate staff with CM Supported for proportion of graduates with PE	Not supported for proportion of post graduate staff Supported for proportion of graduates with PE
H3e The extent of improvement will be positively associated with the proportion of technical staff	Supported for SD; PE	Supported for SD; PE

The research questions and results are discussed in Chapter 8; the conclusions and implications for research and practice are presented in Chapter 9.

7.6 Comparison of best practice adoption with process capability

The objective of this section is to compare the survey data with that collected from the PIP field experiments. As mentioned in §6.14, the two stages of this study enabled triangulation of the survey and PIP results. Table D.3 in Appendix D shows a mapping of best practice survey questions to SPICE/RAPID processes. It is important to note that as the survey only asked was the practice used, it did not measure capability, only the performance (equivalent to capability level 1) of processes. There is another limitation of this comparison: although all assessed organisations were members of SEA Queensland, some of the firms did not respond to the best practice survey. Hence, matched survey responses and RAPID assessments exist for only 16 of the 22 assessed organisations.

As shown in Table 7.50, there were no survey questions relating directly to risk management, or process establishment. There was one question relating to problem resolution and also for requirements elicitation. The quality assurance process included five questions, and there were eight questions relating to configuration management. The processes with the greatest representation in the survey were software development (9 questions), and project management (12 questions).

Table 7.50 Summary of mapping of best practice survey to RAPID processes

RAPID Processes		BP Survey Question Numbers											
RE	Cus.3	1.1											
SD	Eng.1	1.3	1.5	1.6	2.2	2.4	2.5a	2.7	2.9	3.4			
CM	Sup.2	2.6a	2.6b	2.11	3.1	3.2	3.3	3.5	4.4				
PM	Man.2	4.1	4.2	4.3	4.5	4.6	4.7	4.8	4.9	5.1	5.2	5.4	5.6
PR	Sup.8	2.10											
RM	Man.4												
QA	Sup.3	2.5b	5.3a	5.3b	5.7	5.8							
PE	Org.2.1												

To explore the relationship between the survey responses and the assessment findings, firstly, the survey responses for each of the following processes were aggregated to provide a mean process adoption level for each process: software development, configuration management, project management, and quality assurance. SPSS was used to calculate Spearman correlations between the best practice survey adoption level for each process and the process capability level at assessment, but as shown in Appendix I Table I.19, no significant relationships were found between the survey responses and the assessment process capability levels.

To examine the relationship with respect to the attribute achievements, the mean adoption levels for the survey responses relating to software development were compared with the process attribute achievement ratings. Table I.20 in Appendix I lists the mean adoption levels and attribute achievement ratings for all firms, and Table I.21 shows the result of the correlation calculation: there are no significant associations. The same procedure was followed for configuration management (Table I.22 and I.23), project management (Tables I.24 and I.25), and quality assurance (Table I.26 and I.27). The results show that no significant relationships were found between adoption of best practice as recorded by the survey, and process attribute achievement.

Possible reasons for the lack of correlation between the two data collection methods are explored in the next chapter.

7.7 Conclusion

This chapter reported the statistical analysis conducted on the data gathered through the PIP field experiments. The techniques selected to explore the data were identified and explained. Descriptive analysis was used to provide a profile of respondents. Variables were checked for normality to ensure appropriate statistical tests were conducted.

From the initial assessments, it was determined that most of the processes were at low levels of capability: one third of all processes were rated at capability level 0, and almost half the processes were rated at level 1. The attribute achievement ratings of the primary lifecycle processes are higher than that of the support and organisation processes. Organisation and project characteristics were compared against process capability levels and attribute achievement ratings. For many of the processes, higher capability levels are associated with the proportion of experienced staff, the proportion of staff with formal education, the proportion of technical staff, certification to ISO 9001, and firms undertaking projects with longer durations. The performance perceptions regarding budget, schedule and customer satisfaction were found to be associated with some of the process capability levels; and schedule, customer satisfaction, and requirements performance are associated with attribute achievement.

The extent of improvement in process capability was statistically assessed and compared to selected organisational characteristics. Seven firms improved their attribute achievement ratings and six of these firms increased their process capability levels. Overall, the 20 firms achieved a total increase of 26 capability levels. The greatest improvement was recorded for the process establishment process. The proportion of technical staff is associated with the extent of improvement of the software development and process establishment processes. The proportion of staff with formal eduation qualifications is associated with the extent of improvement in configuration management and process establishment.

The best practice survey responses were compared to the RAPID assessment ratings for six of the processes. No significant relationships were found between the responses to the survey questions and the corresponding process capability levels or attribute achievement ratings.

A detailed discussion of the implications of the findings presented in this chapter combined with the survey findings from Chapter 5 is provided in the next chapter.

8 CHAPTER EIGHT—DISCUSSION

8.1 Introduction

In this chapter, the findings from the data analysis of the best practice survey responses (presented in Chapter 5) and the process improvement field experiments (detailed in Chapter 7) are summarised and explained within the context of this and prior research which was examined in Chapter 2. Chapters 5 and 7 focussed on the results of statistical tests, whereas this chapter focuses on the practical importance of the findings.

The aim of this chapter is to discuss the findings in terms of each of the three research questions. As well as discussing the outcomes of the hypothesis tests, this chapter provides context and meaning to the study by comparing the results with similar studies, and by qualitative analysis of the survey responses and PIP reports.

Firstly, the discussion focuses on the findings related to the hypotheses associated with the first research question: is there wide variation in the extent of adoption of software development best practice techniques by software developers in Queensland, and is the extent of adoption related to particular organisational characteristics? Then the second section (§8.3) considers process assessments to compare the capability levels of technical processes with management processes, and discusses the relationship between process capability and organisational characteristics. The third section (§8.4) discusses the findings relating to the third research question: are assessment-based SPI programs an effective means to improve process capability for small software development firms, and is the extent of improvement associated with organisational characteristics? The fourth section (§8.5) uses qualitative analysis to explore issues related specifically to the assessed firms, and also the PIP method. Critical success factors identified in previous SPI studies are then discussed in the context of this study and to highlight factors of particular relevance to small development firms. Following this, section §8.6 discusses the results of the comparison of the data from the best practice survey with the process improvement program, and explores some methodological issues related to the multi-method approach. Finally, the findings are summarised. The next and final chapter provides conclusions, contributions, implications, limitations, and suggests areas requiring further research.

8.2 Discussion of findings related to research question 1

In this section, the findings from the survey data analysis (presented in Chapter 5) are summarised and explained within the context of this and prior research (which was examined in Chapter 2). The first research question explored the extent of adoption of software development best practice, and the relationship between the extent of adoption and particular organisational characteristics.

8.2.1 Variation in adoption of individual practices

As reported in §5.4.1, wide variation was found in the adoption level of the 44 practices covered in the survey: the mean adoption level of the 44 practices is almost 48 percent and the standard deviation is 22 percent.

The wide variation in adoption is evident in Table 8.1 which shows the five least used practices in contrast to the five practices with highest adoption levels. Only 11 percent of organisations gather and analyse test statistics compared to the practice of appointing a project manager for each project, adopted by 87 percent of respondents.

Table 8.1 Comparison of adoption of least and most used practices

Five least used practices	Five most used practices
Statistics on test efficiency are gathered and analysed for all testing stages (11%)	Each software project has a nominated software project manager (87%)
Documented procedure exists for estimating software size and using productivity measures (12%)	Appropriate levels of customer input are made throughout the project (84%)
Records of software size are maintained for each software configuration item and fed back into the estimating process (12%)	Independent testing is conducted by users before any system goes live (80%)
Statistics on the sources of errors in software code are gathered and analysed (18%)	Common coding standards are applied to each software project (80%)
Software tools are used for tracking and reporting the status of the software in the software development library (19%)	Prototyping methods are used in ensuring the requirements elements (78%)

(extracted from Appendix F, Table F.9)

Despite claims that most software development organisations understand the importance of non-technical issues (Dutta, Lee & Van Wassenhove 1999, p. 36), technical processes, such as those relating to customer requirements and software engineering, exhibited higher adoption than non-technical practices such as project management, organisation and support (detailed in §5.4.1).

The adoption levels of practices (as listed in Appendix F Table F.9) are summarised by each practice group in Table 8.2.

Table 8.2 Summary of adoption levels by process groups

ESI practice group	Average adoption	Number of practices adopted in range				
		> 75% of firms	50-75 % of firms	25-50% of firms	<25% of firms	Total no. of practices in group
Standards and procedures	54%	3	6	4	2	15
Control of development process	49%	0	4	2	0	6
Organisational issues	48%	2	2	2	3	9
Tools and technology	47%	1	2	1	2	6
Metrics	35%	1	1	3	3	8
Total practices by adoption range		7	15	12	10	44

Standards and procedures. As shown in Table 8.2, the highest average score, 54 percent, was recorded for the group of questions relating to standards and procedures for estimating, coding, and quality assurance. In analysing the practices that make up the standards and procedures group (as listed in Appendix F, Table F.6), it was found that management plays an active role in most organisations: in 77 percent of firms, a mechanism exists to ensure that projects selected support the business objectives; in almost 70 percent of organisations, management formally assesses the benefits, viability and risk of each project prior to committing to it; in 63 percent of cases, the functionality, strengths and weaknesses of the system being replaced are formally reviewed; and management conducts periodic reviews of the project status in three quarters of the organisations (74%). One of the most widely adopted practices revealed by the survey was the use of coding standards (80%).

Although there was a strong response in regard to users conducting independent testing (80%), and most firms followed a procedure to ensure that the system passing user acceptance testing is the same as that implemented (63%), few follow a procedure to ensure deliverables are understood when handed over from one group to another, such as user to analyst (38%). The high level of adoption regarding user testing is surprising as, in a recent Australian survey about testing practices, Ng et al. (2004) reported that less than half their respondents practiced user acceptance testing. In this Queensland best practice survey, less than half of the respondents commence test planning prior to coding (44%), and independent audits are not conducted at most organisations (35%). Serious weaknesses were identified in regards to estimation: 38 percent use a formal procedure to produce estimates of effort, schedule and cost, and only 12 percent follow a documented procedure for estimating software size or productivity. More than half the respondents did not answer the question about ensuring external software subcontractors followed a disciplined process, but of those who did, 55 percent had such a procedure in place.

Control of the development process. This group of practices, including procedures for controlling changes and testing, averaged 49 percent overall adoption. More than half the respondents had procedures for controlling changes to estimates and schedules (58%), requirements and designs (57%), and code and specifications (59%), but less than half gained agreement from all who contributed estimates and schedules prior to revising such plans (41%). This proportion is much lower than the 77 percent adoption reported by Ng et al. (2004). Although 54 percent of respondents ensured that all functions are tested or verified, only 26 percent use regression testing after initial implementation, surprisingly far lower than the Ng et al. (2004) Australian study which reported 69 percent of respondents performed regression testing. Although recognising that their survey sample might represent better rather than average projects, Cusumano et al. (2003) found very high adoption of regression testing in India (92%) and Japan (96%), compared to 71 percent in the United States and 77 percent in Europe.

Organisational issues. Responses averaged 48 percent for questions relating to organisational structure and management practices, lines of authority and resource availability. Most organisations have nominated a project manager for each project (87%), but in almost half the cases, the software manager does not report to a business project manager (44%).

The importance of customer and marketing input is widely recognised (84%), and more than half the organisations have a procedure to ensure essential non-software resources are available (61%). Less than half the respondents establish a change control function for each project (45%) and only 23 percent have a software quality assurance function, far lower than the 50 percent reported from a recent survey of Northern Ireland software developers (McCaffery et al. 2004). Practices in need of improvement include training new software managers in the firm's procedures, and also maintaining awareness of software engineering technology (both 21%).

Tools and technology. An overall adoption of 47 percent was recorded for practices pertaining to the use of tools and technology for management and development activities. Prototyping methods are widely used to ensure requirements (78%), and data dictionaries are used by 69 percent of respondents. Detailed design techniques (such as SADT) are used by half the organisations surveyed (51%), a much higher proportion than that reported recently in Northern Ireland (18%) by McCaffery et al. (2004). The use of detailed design showed wide variation in adoption in a survey reported by Cusumano et al. (2003): India 100 percent, Japan 85 percent, United States 32 percent, Europe and others 68 percent. The explanation for the variation was that developers in the United States were following the practice of Microsoft programmers who go from functional specification to code to avoid wasting time documenting features which may later be deleted (Cusumano et al. 2003).

Despite the availability of software such as MS Project, less than half the respondents (43%) use project management software for planning, estimating and scheduling. There is very low use of software tools for tracing requirements (20%), or tracking and reporting the status of software in the software development library (19%). More detailed analysis of the use of programming languages and development tools was provided in §5.3.6, indicating a trend towards Java, and unexpectedly, a decrease in all tools except source code control (up 1%); automated test capture (up 3%) and code coverage (up 1%).

Metrics. The group of practices exhibiting the poorest average rate of adoption (34%) covered the gathering and use of statistics to improve error prevention and project estimation. In general, tracking and performance metrics are used more widely than quality and estimation metrics. Post-implementation problem reports are logged and tracked by 76 percent of firms, and records exist for 63 percent of respondents to reconstruct current versions of software. Only one third of firms compare actual project resources with estimates, use earned value to monitor progress, or compare actual with estimated system performance. To add to this concern, as mentioned in §5.3.6, the proportion of firms using staff time sheet systems is expected to drop from 34 percent to 19 percent, thereby reducing the opportunity to capture measures of actual effort. Root cause analysis is performed by only 18 percent of respondents, and only 11 percent of organisations gather and analyse test efficiency.

Summary

In summary, the respondents used project management estimation but generally did not monitor the actual use of resources. The relatively high levels of management involvement indicated commitment and involvement of management, possibly due to the large number of small firms. Most organisations appear to have a healthy relationship with their customers, involving them in requirements management and user acceptance, and also effectively logging problems and tracking issue resolution. However, the use of metrics and tools is low, restricting full understanding of the development process, and limiting productivity.

8.2.2 Variation in adoption across software development industry

As well as the wide variation in terms of the adoption of individual practices (§8.2.1), there is wide variation in the overall adoption of best practice by organisations: as reported in §5.4.2, responding organisations have adopted, on average just less than half (48%) of the software best practices put forth in the survey. The standard deviation of 21 percent indicates wide variation across organisations in the level of adoption of the 44 practices. It reflects poorly on the software development industry in Queensland that, as shown in §5.4.2, 16 percent of organisations have adopted less than 25 percent of the practices in the survey. On the other hand, 10 percent of respondents believe such practices to be worthwhile and claim to be using at least three quarters of the practices in the questionnaire.

Characterising the high adopters from the low adopter organisations

To provide guidance to software engineering managers, and possibly purchasers, Dutta, Lee and Van Wassenhove (1999) used discriminant analysis to identify the practices which distinguish the high level adopters from the low level adopters, and found the following four practices accounted for the greatest variance:

- Establishing a change control function for each project
- Maintaining awareness of the state of the art in software engineering technology
- Collecting and analysing statistics on sources of errors in code
- Prototyping to meet requirements (1999, p. 88).

The practice of using comparisons between two extremes for drawing inferences is used by other researchers (O'Regan & Ghobadian 2004). Although the discriminant analysis requires that the variables are continuous and normally distributed, neither condition met by the Queensland survey data, it was decided to attempt the analysis in order to compare the Queensland survey results with the ESI results. As discriminant analysis cannot be calculated for variables with missing values, seven of the 44 questions which had a high proportion (>20%) of *not applicable* or missing values were excluded from this analysis (questions excluded: Q1.2; Q1.4; Q2.5b; Q2.8; Q4.9; Q5.3b; Q5.5).

The 203 responses were sorted by overall adoption level to identify the organisations with high overall adoption (coded as 1) and also the organisations with lowest adoption (coded as 0). As the test excludes cases with missing values it was necessary to repeatedly run the discriminant analysis test selecting additional cases with high and low overall adoption until 20 complete cases were selected in each category. From these 40 cases, the discriminant analysis identified only one question accounting for the variance between high and low overall adoption: Q4.2—*Use of a formal procedure to produce software development effort, schedule, and cost estimates* (Wilks Lambda=.001, test results in Appendix J, Table J.1). This outcome is consistent with Jones' view that poor estimating and planning are consistently characteristics of major software disasters (Jones 1996).

However, due to the fact that the proscriptions of the discriminant analysis test had been violated, and the responses (as shown in Appendix J Table J.2) indicated that many of the best practice questions were answered *yes* by the high adoption organisations and *no* by the low adopters, it was decided to conduct Fisher's exact test to compare the responses of the 20 high level adopters and 20 low adopter organisations for each question. For the Fisher's exact tests, all 44 questions were included, and the cases were selected based solely on their overall adoption level (unlike in the discriminant analysis where cases with missing values were excluded). The results of the Fisher's exact tests are summarised in Appendix J, Table J.2. It is recognised that some of the best practice questions are inter-related to some degree, so to take this into account, the p value was multiplied by the number of questions (44) (Bonferroni correction as advised by Miller 2004, p. 184).

The Fisher's exact test found that a significant difference exists between the high adoption group and the low adoption group for 40 of the 44 survey questions. The only questions where there was not a significant variation were:

- Q1.5 Prototyping methods used in ensuring the requirements elements of the software
- Q2.4 Common coding standards applied to each software project
- Q5.1 Each software project has a nominated software project manager
- Q5.2 The software project manager reports to a business project manager responsible for the overall benefit of the project to the business.

It appears from this analysis that the variation in practice between the high and low adoption organisations as reported by the ESI is not as great as the variation in the practice adoption of the Queensland organisations, although it is recognised that the analysis techniques differed. This analysis provides further support for the hypothesis that a substantial variation exists in the adoption of best practice (because for 40 of the 44 practices, the adoption level of the 20 highest organisations is significantly higher compared to the adoption level of the 20 lowest organisations). This finding supports the conclusion reached by Ng et al. (2004) in their empirical study of software testing practices in Australia which revealed 'the existence of some degree of inconsistencies and weaknesses within the software development practices in industry' (pp. 119-20).

Having established that wide variation in overall adoption exists across the organisations, the next four hypotheses considered organisational characteristics thought to be associated with adoption: organisation size, software development team size, type of software development undertaken, and industry sector.

8.2.3 Organisation size

As reported in §5.4.3, the association between the total number of employees in the organisation and the adoption level of best practice is not statistically significant for COTS or non-COTS developers. Due to overlapping responses to the question relating to primary involvement in the software industry, it was not possible to separate responses from in-house developers from those of commercial developers. However, a review of the names of the medium and large organisations revealed that most of them were not commercial software development organisations. When the organisations that recorded their primary involvement in the software industry as software user (either developed in-house or by a 3rd party) were excluded, then a strong correlation was found for organisation size and adoption of best practice for the remaining non-COTS (p=.042) and COTS (p=.002) organisations (statistical results in Appendix F Table F.17).

This suggests that in-house development groups have poorer levels of best practice compared to commercial software development firms. This result is not surprising as organisations primarily using 3rd party developed software would not need to implement many of the development best practices included in the questionnaire as they would be more concerned with evaluating, purchasing, installing and supporting packages. Also, in very large organisations, the in-house development group is a relatively small part of the organisation and may be focussed on maintenance and enhancement of systems rather than development. In-house developers in such organisations may be motivated by the corporate goals rather than software development best practice, and may not experience the competitive necessity of best practice adoption, a view shared by McCaffery et al. (2004): ‘for organisations where software development is not their main business, software processes are never going to be an issue’ (p. 160). Furthermore, it has been claimed (O’Regan & Ghobadian 2004) that larger firms put more effort than small firms into achieving exceptional performance because they tend to 'compete directly with other firms in the market place rather than supply one or more larger firms' (p. 74).

From a survey of software development groups in Northern Ireland, McCaffery et al. (2004) noted that larger companies and companies with mandatory standards place more emphasis on adhering to quality standards and procedures than smaller companies. They posit this is because large software development companies need processes to cope with complexities of managing large numbers of employees, but small companies can do without such processes as they have less complex and more straightforward communication. This view is reinforced by O'Regan and Ghobadian (2004) who noted that firms with less than 20 employees emphasised freedom for employees to depart from the rules to a greater extent compared with firms with in excess of 100 employees.

8.2.4 Software development team size

As shown in §5.4.5, the size of the development group is significantly associated with the extent of adoption of best practice, a finding consistent with prior studies. For example, the ESI reported a 'monotonic rise' (ESI 1995, p. 31) in average adoption of software organisations grouped by size (based on the number of software employees). Also, in New Zealand, Groves et al. (2000) observed larger development groups tend to have more well-defined software development processes.

8.2.5 Development type COTS/nonCOTS

The large proportion of respondents who develop commercial off-the-shelf software was surprising, but, according to the CEO of SEA (Qld) (at the time of the survey), Phil Scanlan (1999, pers. comm., 04 Jan), the large number of developers (88) who saw their primary role as COTS developers reflects the large concentration of vertical niche market package developers in the Brisbane area. SEA (Qld) provided incubator and language translation services to small firms at its premises in Brisbane. Therefore, SEA (Qld) had a high profile with this community of developers, who in turn showed their support for SEA (Qld) by completing and returning the survey.

The large difference in adoption levels between the two groups of developers begs the question: are there specific practices or groups of practices which are more readily adopted by one group of developers compared to the other? When adoption levels were analysed on a question by question basis, it was found that for the following five practices, COTS developers reported adoption levels at least 16 percent higher than the non-COTS developers:

- Maintain records from which all current versions and variants of software systems and their components can be quickly and accurately reconstructed in the development environment (Q2.11 COTS 24% higher than non-COTS)
- Establish a change control function for each software project (Q3.1 21%)
- Log, track and analyse post-implementation software problem reports (Q2.10 21%)
- Apply common coding standards to each software project (Q2.4 17%)
- Use software tools used for tracking and reporting the status of the software subroutines in the software development library (Q3.5 17%).

Comparing the adoption of non-COTS and COTS developers for each question revealed that for 10 of the 44 questions, COTS developers showed significantly higher adoption than non-COTS developers (full list of results in Appendix J Table J.3). In fact, there were only two practices, both related to project management estimation, where non-COTS developers exhibited higher adoption compared to the COTS developers, but the difference was not statistically significant. Non-COTS developers reported six percent higher adoption of the use of formal procedures to produce software development effort, schedule, and cost estimates; and four percent higher adoption for control of estimates, schedules and subsequent changes only by the project managers.

As mentioned in §8.2, the ESI report (1997) groups the practices under five headings: organisational issues; standards and processes; metrics; control of the development process; and tools and technology. As can be seen from the data provided in Table 8.3, COTS developers show higher adoption levels for every group of processes compared to non-COTS developers, and in fact show higher adoption than the ESI respondents for standards and processes, and tools and technology. The ESI researchers did not report separate analyses for COTS/non-COTS groups.

Table 8.3 Comparison of adoption by ESI groups for non-COTS and COTS organisations

Practice group	Average adoption levels			
	ESI[a]	Non-COTS[b]	COTS[b]	Difference COTS – non-COTS
Organisational issues	58%	44%	53%	9%
Standards and processes	51%	54%	59%	6%
Metrics	45%	31%	42%	11%
Control of development process	58%	47%	52%	5%
Tools and technology	45%	43%	52%	9%

(Source: a ESI 1997, b from this study)

To gain an understanding of which software engineering practices are most used in Queensland, the highest scoring questions were collated and ranked for COTS and non-COTS developers. This comparison is presented in Table 8.4.

Table 8.4 Most used software engineering practices

COTS rank	Practice	Non-COTS rank
1	Each software project has a nominated project manager (Q5.1)	1
2	Common coding standards are applied to each project (Q2.4)	6
3	Post-implementation problems are logged and their resolution tracked and analysed (Q2.10)	8
4	Appropriate level of user/customer/ marketing input is made throughout the project (Q1.1)	2
5	Independent testing is conducted by users under SQA (Q2.5a)	3

Of the five most used practices, three relate to coding, testing and problem resolution, the other two involve requirements and project management. The popularity of these practices is not surprising, as the importance of these techniques has been recognised in the industry press and stressed in information systems and software engineering training courses for some time.

The higher adoption level exhibited by COTS developers raises the question: why have COTS developers adopted best practice techniques at higher levels compared to their non-COTS counterparts? One reason could be that, due to the limited size of the Australian domestic market, COTS developers are competing in the global marketplace. In order to be commercially successful, these developers may have realised that their software processes must be of a high standard to respond quickly with a quality software product. In contrast, the overall shortage of information technology staff leading up to the time of the survey may have caused complacency amongst in-house and custom developers. This staff shortage offers a sense of job and contract security; clients may be prepared to suffer poor practices rather than risk delayed development.

Another reason could be related to the length of time the COTS organisations have been in existence. It is likely that most COTS developers would be established more recently than in-house and custom developers, and would have commenced operation with current best practice (Eisenhardt 1988). It may be that staff in the COTS firms are more experienced developers and therefore more aware of the benefits of new techniques (Leonard-Barton 1987).

So, while results of previous research in this area have been inconclusive, Bawden's (1994) emphasis on the value of reputation may suggest a solution: long-established non-COTS developers may have gained the trust of their clients, enabling them to compete on reputation and past performance, without best practices. In contrast, newer COTS development firms may see adoption of best practice as a competitive necessity to break into the global market.

8.2.6 Industry sector

Other studies claimed that some sectors are characterised by higher adoption, for example, application domains such as aerospace, banking, process control, productivity tools and reservation systems (Dutta, Lee & Van Wassenhove 1999; Glass 1996). This study reported here found that significant variation existed across the various sectors represented by non-COTS developers (reported in §5.4.6), with higher adoption recorded for organisations representing the finance and insurance sector, and the utilities sector compared to organisations from the mining or education sectors. Furthermore, organisations from the government administration and defence sector, software development sector and information technology sector were all significantly higher compared to those from the education sector.

It came as no surprise that most of the COTS developers recorded their sector as software development, and there were few responses (only 9 of 87) in the other sector categories. Overall, 77 percent of respondents recorded their sector as software development. This is in contrast to the responses to the three ESI surveys; software consultancy and supply accounted for about 40 percent of responses (Dutta, Lee & Van Wassenhove 1998).

8.2.7 Comparison of best practice in Queensland with European countries

The mean overall adoption level of the responses from Queensland organisations of 48 percent with a standard deviation of 21 percent is slightly lower than the average reported by the ESI from its 1997 questionnaire (51%±21) (ESI 1997). However, the ESI acknowledges that if their respondents had not been self-selected for their interest in software process improvement, the ratings would be lower than those reported (Dutta, Lee & Van Wassenhove 1999).

As can be seen from the adoption level breakdown by country (Table 8.5) for the last ESI survey, best practice adoption in Queensland rates higher than adoption in a number of European countries.

Table 8.5 Overall responses and average adoption level by country

Country	No. of responses	Average adoption level
France	18	65%
United Kingdom	52	60%
Greece	18	57%
Denmark	17	55%
Finland	4	55%
Norway	6	53%
Austria	16	53%
Italy	77	52%
Queensland COTS developers [a]	**87**	**52%**
Netherlands	30	49%
Germany	62	49%
Queensland overall adoption[a]	**203**	**48%**
Israel	11	46%
Ireland	12	45%
Queensland Non-COTS developers [a]	**116**	**44%**
Spain	34	44%
Belgium	15	43%
Sweden	13	32%

Note: [a]Queensland results from this study; European results from survey of year 1997 (ESI 1997).

However, caution must be exercised in comparing the results for the Queensland survey with the ESI survey as the ESI survey was applied in a different context. The

ESI survey was completed by firms seeking EC funding, whereas the survey reported here attempted to conduct a census of all organisations involved in software development in Queensland. Further results available from ESI are provided in Table 8.6 to compare the leaders and laggards in best practice adoption from the ESI surveys over three years from 1995 to 1997.

Table 8.6 European results: leaders and laggards

Year and no. of responses	Highest adoption		Lowest Adoption	
	Country	Adoption level	Country	Adoption level
1995[a] 425 responses	United Kingdom 29 responses	65%	Sweden 8 responses	38%
1996[b] 457 responses	France 20 responses	68%	Spain 65 responses	37%
1997[c] 397 responses	France 18 responses	65%	Sweden 13 responses	32%

(Source: a Dutta, Van Wassenhove & Kulandaiswamy 1998);(c.Dutta, Lee & Van Wassenhove 1999; b.ESI 1996b)

However, the difference in the adoption levels across Europe raises the question of national cultural issues, which has been briefly explored by Dutta, Lee and Van Wassenhove (1998) who used Ronen and Shenkar's (1985) national culture clusters to compare adoption of clustered countries. Dutta, Lee and Van Wassenhove (1998) observed that Germany and Austria behaved similarly; however, with respect to Scandinavian countries, they found considerable variance warranting further research. The clusters derived by Ronen and Shenkar measure work goals, values, needs, and job attitudes and are named Anglo, Germanic, Nordic, Latin European, Latin American, with Australia classed in the Anglo cluster along with United Kingdom, Ireland, USA, Canada, New Zealand and South Africa (Mahoney et al. 2001, p. 405).

Recently, other researchers have explored the role of national culture in software development. For example, substantial cultural factors were identified by Paulish and Carleton (1994) as evidenced by differences in adoption of software process methods in Siemens sites in Germany and USA. In commenting on the fact that CMMI and SPICE have two dimensions, the process dimension and capability dimension, Biro, Messnarz and Davison (2002) call for a third dimension to CMMI and SPICE, the *cultural* dimension because 'the national cultural position of the company may determine a different meaning and suitable improvement actions' (p. 36) for every process-capability pair. Biro, Messnarz and Davison (2002) refer to Hofstede's five generic factors that characterise value systems in different national cultures: power orientation; individualism versus collectivism; masculinity versus femininity; uncertainty avoidance; and long term versus short-term orientation. These generic factors are illustrated in Figure 8-1.

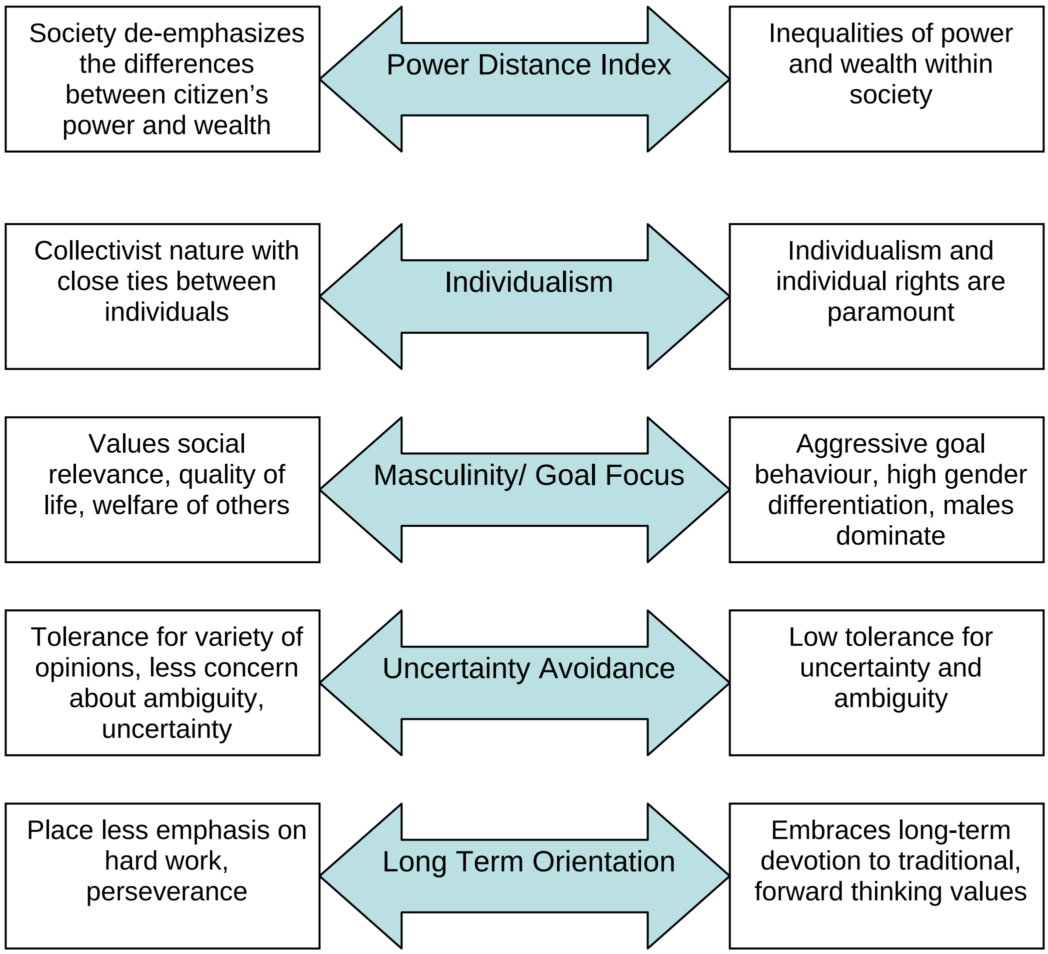


Figure 8-1 Hofstede's five generic cultural factors
(adapted from Hofstede; Mahoney et al. 2001).

Other researchers have applied Hofstede's scores in IT related research, for example, Frank et al. (2001) found evidence that innovativeness correlates with the tendency for uncertainty avoidance in a study of the adoption of mobile technology across Finland, Germany and Greece; and more recently, Borchers (2003) applied Hofstede's theory to understand project problems experienced by project teams of Indian, Japanese and American software developers.

In relating Hofstede's dimensions to the adoption of best practice techniques, it could be expected that higher adoption may be associated with low uncertainty avoidance (willingness to adopt new techniques), and low individualism (conformance to group working practices). Hofstede's scores indicate that Australians, compared to others, have low uncertainty avoidance (would be quick to adopt innovations) but high individualism (resistant to standard work practices) (Mahoney et al. 2001). In Appendix J, Table J.4, Hofstede's scores are recorded against the adoption levels from the ESI survey, and also the Queensland survey responses. Pearson's correlation test found no significant correlation of best practice adoption with any of Hofstede's dimensions (result in Appendix J, Table J.5).

As well as taking into account the variation in national cultures, Krutchen (2004) believes 'everyone knows the difficulty of adapting technology and methods to other cultures' so the variation in adoption across different countries should come as no surprise. Krutchen (2004) also highlights the range of practices which are candidates to be classified as *best practice*. The issue of defining best practice is addressed in the next section.

8.2.8 Compilation of list of best practice techniques

As stated in §2.4, the content of the questionnaire has been criticised for not covering all possible best practice (Dutta, Lee & Van Wassenhove 1999), specifically that it overlooks important issues related to organisational and customer-supplier management; and it does not including high level practices (Dutta & Van Wassenhove 1997a). When the best practice questionnaire was mapped to the ISO 15504 processes (Appendix D Table D.3), it became apparent that many of the ISO 15504 processes were not included in the best practice questionnaire. Considering the rigour and effort invested in the formulation of ISO 15504, these processes must be considered to be important, to be *best practice*. As shown in Appendix J, Table J.6, the best practice questionnaire does not cover 13 of the 24 ISO 15504 base processes. The best practice questionnaire heavily emphasises project management, but has no practices relating to risk management, measurement, validation, joint review or audit. There is only one question in the best practice survey which is not covered in ISO 15504: procedures to ensure that the functionality, strengths, and weaknesses of the system which the software is replacing are formally reviewed (Q1.4).

As world class standards are dynamic, best practices may be temporary (Cortada 1997), especially in software engineering which has frequently adapted to changes brought about by evolution of technology (Finkelstein & Kramer 2000). With the passing of time, best practice becomes standard practice as other superior practices emerge (Cragg 2002). Recently, agile software engineering methods have been promoted as best practice. Proponents of agile methods would compile a different set of best practice techniques, focussing on customer satisfaction and early incremental delivery of software; small highly motivated project teams; informal methods; minimal software engineering work products; overall development simplicity stressing delivery over analysis and design; and active and continuous communication between developers and customers (Pressman 2003).

Prior to the survey execution, one of the members of the SEA National Board questioned the relevance of the techniques in the survey to COTS developers (B Hoff 1998, pers. comm., 30 November). Mr Hoff raised the point that Australian COTS developers may be using excellent practices which are not included in the ESI questionnaire, for example, management of Beta tests. Jones (2003) notes that Beta testing has been used since the 1960s, and Cusumano et al. (2003) reported its widespread use at 73 percent. Therefore, it is recognised that the items from the ESI questionnaire may not provide an entirely valid measurement of best practice across the industry. This point is acknowledged by the ESI: 'progress in software engineering may not be visible along dimensions measured in the survey' (ESI 1998, p. 29).

Another issue to consider is that software practices may have changed significantly in the six years between the design of the ESI questionnaire and its use in the Queensland survey. For example, reuse is now recognised as one of the most valuable software engineering practices (Mili, Mili & Mili 1995) but is not included in the ESI questionnaire. So while longitudinal studies such as that undertaken by the ESI are valuable in mapping the take-up rate of recommended techniques and practices, the data collection tools need to be kept up-to-date while still providing comparative data.

A further issue concerns the interpretation of the questions. Many of the questions were worded 'does a procedure exist …' and some respondents may have interpreted this in different ways. For example, is there a documented procedure? If so, is it used in all projects? Some may have responded *yes* if they occasionally perform the practice, although the procedure may not be documented at all. Variation in adoption may depend on the individual developer as well as the project, although Henninger, Sieber and Dilger (2001) note that developers' individual working styles become homogenised into the 'shop floor' culture and cite Strübing's study which showed that there is a clear inconsistency between what developers *think* should be done and what they really *do*. Inconsistencies such as those noted by Neill and Laplante (2003) could occur due to variations in interpretation of survey questions, or developers genuinely mixing disparate techniques, such as their finding that projects used prototyping within a waterfall methodology, and use cases without object oriented design.

8.2.9 Summary of discussion relating to research question 1

The survey responses, representing over 200 organisations, indicate that Queensland organisations involved in software development have adopted, on average, almost half of the best practices. The survey results indicated that wide variation exists in the adoption of best practice. Firstly, some of the practices, especially those relating to project management planning and customer involvement are widely adopted, but others, in particular the use of metrics for estimating and testing, are barely used by the organisations which responded to the survey. Generally, practices of a technical nature are more widely adopted compared to techniques related to support and management. Secondly, considering the level of adoption by organisations, wide variation was recorded as well. For 40 of the 44 practices, the differences between the 20 high adopters and 20 low adopters were significant, indicating a large difference in the work practices of the leaders compared to the laggards.

Analysis of organisation adoption level revealed the responses represented two distinct populations: organisations developing commercial off-the-shelf systems exhibited higher adoption compared to those which do not develop COTS systems. For about one quarter of the practices examined, COTS developers are significantly better than non-COTS developers. Higher adoption is associated with the size of the development team, highlighting the need for small teams to enhance their skills, and to assess and improve the processes currently used.

Therefore, there is wide variation in best practice adoption in terms of the adoption of individual practices, and also in terms of organisation adoption levels. Furthermore, adoption is associated with large development teams, and is more advanced in organisations which develop COTS systems. For those not developing COTS systems, the organisations from the finance and insurance, and utilities sectors exhibited highest adoption of best practice, with mining and education the lowest. The aspects of software development with most scope for improvement include technical issues (specifically relating to testing), the need for automated support (especially for configuration management), and project management and quality assurance.

This part of the study has provided a snapshot of the current state of Queensland developers, and in comparing the results from this survey with earlier results from Europeans countries, shows that Queensland organisations are comparable to European firms (especially the COTS developers). However, the need to update the questionnaire was highlighted to include additional techniques which are now widely accepted as best practice.

The survey provided an opportunity for organisations to compare their practice adoption with the average from all the other respondents, and provided insights into possible improvement paths for organisations, researchers, and software industry bodies such as SEA (Qld). In Northern Ireland, over half the software developers expressed a desire for some form of improvement and 23 percent were in favour of CMMI or SPICE (McCaffery et al. 2004). As detailed in §5.5, 51 percent of Queensland respondents provided details of desired improvements and 35 percent suggested how SEA (Qld) could help them to improve their processes. The next section presents the findings from the SEA-sponsored process improvement program involving 22 small software development firms.

8.3 Discussion of findings related to research question 2

In this section, the findings from the analysis of the process assessment reports of 22 firms (presented in Chapter 7) are summarised and explained within the context of this and prior research (which was examined in Chapter 2). The second research question asked: is the capability of technical processes higher than that of management processes, and is process capability associated with particular organisational characteristics?

As explained in §7.3.1, a total of 176 process ratings were recorded during the initial assessments, with eight processes assessed for each of the 22 firms. Most of the processes were rated at the lowest levels. Almost one third of all the processes were rated as *incomplete* (level 0) and 46 percent were rated as *performed* (level 1). As shown in Figure 8-2, this group of 22 firms exhibited lower capability in comparison to both the Australian and international participants in phase 2 of the SPICE trials (Jung et al. 2001; Rout, Tuffley & Hodgen 1998).

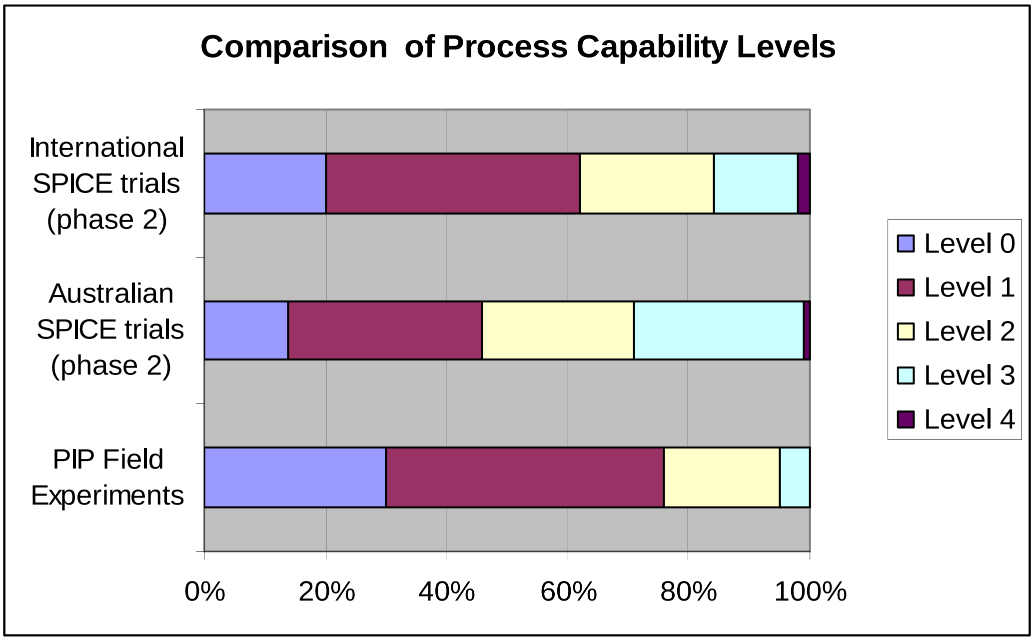


Figure 8-2 Comparison of capability levels PIP and SPICE trials phase 2
(Source: PIP field experiments and Jung et al. 2001; Rout, Tuffley & Hodgen 1998)

8.3.1 Primary and supporting processes

As predicted (in §2.7.2), technical processes such as requirements elicitation and software development were rated higher in capability than the supporting processes (project management, configuration management, problem resolution, quality assurance, risk management, and process establishment). It may be that managers of the firms agreed with Damanpour and Evan's (1984) view that technical innovations are more observable, have higher trialability, and are perceived to be relatively more advantageous than administrative innovations. However, the results from the PIP analysis contrasts with that from rapid SPICE-based assessments on 36 SMEs in regional Italy, which revealed a different result: 'generally, companies obtain better evaluations in organisational than in technological aspects' (Bucci, Campanai & Cignoni 2001, p. 38). Bucci, Campanai and Cignoni found that organisational processes averaged at capability level 2.6, while the average capability level for technical processes was 1.75. However, these researchers used a different method of categorising processes as organisational or technical and commented that the results indicated poor technology in support of process and quality control rather than poor technology in general.

8.3.2 Organisation and project characteristics

Detailed results from the hypothesis tests relating to research question 2 were presented in §6.3, and the summary of the relationships between the organisation and project characteristics with both the capability levels and attribute achievement ratings for the eight processes is reproduced here in Table 8.7 for the convenience of the reader.

As shown in Table 8.7, for all eight processes, capability is associated with at least one of the organisation or project characteristics. The processes showing greatest correlation are quality assurance and project management: these are positively associated with proportions of experienced staff, technical staff, ISO 9001 certification, and firms with projects of longer durations. Some of the organisation and project characteristics are not associated with capability level of any processes, for example, size in terms of number of employees, age of firm, and average percentage cost overrun were not associated with the capability levels of any of the eight processes.

Table 8.7 Summary of significant correlations of organisation and project characteristics with process capability

Organisation/project characteristic	N	RE	SD	CM	QA	PR	PM	RM	PE
Size – headcount	18								
Proportion of experienced staff	21		XY	XY	XY	XY	XY		
Proportion of post graduate staff	22	XY			XY				Y
Proportion of technical staff	22		XY		XY		XY		X
Age of firm	18								
Target business sector – public sector clients	22							Y	
ISO 9001	22	XY	X		XY		X	XY	XY
Projects in progress (-ve association)	21			X					
Number of staff/project	21						Y		
Project duration	21	Y	XY	Y	XY	XY	XY		Y
Cost overrun	13								

X indicates that a significant relationship exists for capability level
Y indicates that a significant relationship exists for attribute achievement

Size factors

Although the firm size in terms of number of staff was not associated with process capability, the size of the project team was associated with attribute achievement of the project management process. However, the number of projects in progress could also be used as a measure of firm size, but showed a negative association with the capability level of the configuration management process.

In the SPICE phase 2 trials, organisation size was defined as a binary variable: small organisations having 50 or less IT staff, and large with more than 50 IT staff (Jung & Hunter 2001). In the SPICE trials, the difference in capability between the 206 process instances from small organisations and the 446 process instances from large organisations was significant for two processes: maintain system and software, and human resources management. To perform a similar comparison with the PIP assessments, the capability levels of the three firms with staff in excess of 50 were compared with the 19 smaller firms. The Mann-Whitney U test revealed that capability of the quality assurance process is associated with size (summary results in Table 8.8, details in Appendix J Table J.7).

Table 8.8 Comparison of process capability for small and large firms

Capability levels	RE	SD	CM	QA	PR	PM	RM	PE
Mann-Whitney U	19.00	18.00	24.50	8.00	25.50	16.00	19.00	19.00
Wilcoxon W	209.00	208.00	30.50	198.00	215.50	206.00	209.00	209.00
Z	-.993	-1.162	-.432	-2.229	-.336	-1.297	-1.011	-1.109
p (2-tailed)	.321	.245	.666	.026(*)	.737	.195	.312	.268
Attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
Mann-Whitney U	17.00	18.00	27.50	6.00	26.00	15.50	15.50	13.50
Wilcoxon W	207.00	208.00	217.50	196.00	216.00	205.50	205.50	203.50
Z	-1.106	-1.014	-.096	-2.189	-.240	-1.250	-1.253	-1.464
p (2-tailed)	.269	.311	.923	.029(*)	.810	.211	.210	.143

Proportion of experienced staff

For both capability measures (capability level and attribute achievement), the proportion of staff with at least five years experience is positively associated with the capability of software development, configuration management, quality assurance, problem resolution and project management processes. This is consistent with the findings of Mustonen-Ollila and Lyytinen's (2003) study of adoption of IS process innovation (based on Roger's diffusion of innovation theory). They found that past technological experience was one of the most important factors affecting the adoption of information systems process innovations. Also, Leonard-Barton (1987) found that staff experience was associated with the adoption of new processes (mentioned previously in §8.2).

It is acknowledged that the measure of proportion of experienced staff recorded on the demographic questionnaire is rudimentary in that the sponsor recorded how many staff had more than five years experience, and how many had less than five years experience. Furthermore, the question did not specify the type of experience involved. However, it was surprising that risk management was not associated with staff experience, as firms with higher proportions of staff with at least five years industry experience should be more aware of the variety and potential impact of risks compared to the less experienced developers, who may tend to be more optimistic. As far as process establishment, it may be the case that the more experienced developers have set ways of doing things and do not see the need to have processes to establish and maintain process assets.

Proportion of graduates and technical staff

Two measures of formal education were derived from the survey (defined in §6.9.2): proportion of staff with graduate qualifications; and proportion of staff with post graduate qualifications. For capability level and attribute achievement the proportion of post graduate staff is associated with capability of the requirements elicitation and quality assurance processes. In addition, attribute achievement of the process establishment process is associated with the proportion of post graduate staff.

For both capability level and attribute achievement, a higher proportion of technical staff is associated with higher capability in processes for software development, quality assurance, project management and process establishment.

Further analysis revealed a strong correlation between the proportion of technical staff and the proportion of graduate staff (Pearson correlation coefficient=.589, p=.002, 1-tailed test). This result is not surprising as most permanent staff would be classified as either technical or administrative by the Federal or Queensland State industrial employment award, with employees in the technical stream required to hold relevant qualifications (QPS).

Duration of firm operation and project duration

Unlike the findings from the Northern Ireland survey (McCaffery et al. 2004), in this study, the age of the firm was not associated with capability for any of the eight processes. However, duration of projects is associated with all processes with the exception of risk management. The strong link between project duration and process capability can be explained in that longer projects are usually more complex and costly and therefore need formalised processes and more sophisticated coordination and management (Schwalbe 2003). Neil and Laplante (2003) found from an analysis of 194 survey responses relating to requirements engineering, that 60 percent of projects lasting less than one year keep to schedule and budget, compared to 20 percent of projects that run longer than two years. Therefore longer projects are more risky, and firms undertaking projects of greater duration are well advised to improve their risk management processes.

Target business sector

Although no single target sector was associated with higher capability, it was found that firms targeting public sector clients such as public utilities, public administration, defence and telecommunications sectors had significantly higher attribute achievement ratings compared to the group of firms which did not target any publicly funded sectors. The higher capability of firms targeting public sector clients is a response to more stringent acquisition policies and procedures enforced by public sector organisations, for example, from 1992 to 1996, the Queensland government had a mandatory purchasing policy in place for ISO 9000 certification (Gome 1996b).

ISO 9001 certification

Only 2 of the 22 firms held ISO 9001 certification, and they have significantly higher capability levels for six of the eight processes: requirements elicitation, software development, project management, risk management, quality assurance and process establishment. However, caution should be exercised in interpreting the result of the statistical test due to the small proportion of certified firms. Nonetheless, this finding is consistent with that from the SPICE phase 2 trials as reported by Jung and Hunter (2001). In the SPICE trials, data from 70 assessments involving 44 organisations were used to compare the capability of 350 process instances from ISO 9001 organisations with the capability of 341 process instances of non-ISO 9001 organisations. Jung and Hunter (2001) report that for 10 of the 29 processes, the capability level of ISO 9001 organisations is significantly greater than that of the non-ISO 9001 certified organisations (p=.05) (p.52).

The SPICE phase 2 trials were based on the 1996 version of the standard (ISO/IEC PDTR 15504 1996) whereas the PIP assessments were based on the 1998 version (ISO/IEC TR 15504 1998). The structure of the process dimension changed from the 1996 version to the 1998 version, and some of the processes names were modified. Seven of the 10 processes identified by Jung and Hunter (2001) are included in the PIP study. For four of these processes, capability is significantly associated with certification to ISO 9001 in this study: software development, project management, quality assurance and process establishment. Two of the processes identified in the SPICE phase 2 trials as associated with certification are not significantly associated in this study: configuration management and problem resolution.

Included in the SPICE phase 2 trials were 15 Australian organisations, 14 of which were certified to ISO 9001 (Rout, Tuffley & Hodgen 1998). The PIP firms reported in this study varied in that regard: only two of the firms in this group of 22 responded that they had ISO 9001 certification, although one other was in the process of gaining certification. It is not surprising that higher capability would be associated with ISO 9001 as a comprehensive mapping by Hailey (1998) determined that all 200 base practices of ISO/IEC PDTR 15504 corresponded to at least one of the 20 clauses of ISO 9001-3. Although certification to ISO 9001 is associated with higher process capability, the lack of ISO 9001 should not be assumed to be an indicator of low capability as firms with high capability may not have ISO 9001 certification ‘because it is not necessary for their business’ (Jung et al. 2001, p. 234). However, due to the similarities between ISO 15504 and ISO 9001, firms with high process capability would find it relatively easy to achieve ISO 9001 compared to low capability firms.

8.3.3 Perceptions relating to performance

Included in the questionnaire completed prior to the assessment were questions about the perceptions of the assessment sponsor in relation to the firm’s ability to meet budget and schedule commitments, to achieve high customer satisfaction, to meet specified and implied requirements, staff productivity, and staff morale and job satisfaction. As explained in §7.3.11, these perceptions were compared with the capability levels of the eight processes. A summary of the results of the correlation tests is provided as Table 7.34 and reproduced here in as Table 8.9 for the convenience of the reader.

Table 8.9 Summary of associations between capability levels and perceptions about performance indicators.

Perceptions relating to firm’s …	N	RE	SD	CM	QA	PR	PM	RM	PE
ability to meet budget commitments	19						X		
ability to meet schedule commitments	22				Y		XY		XY
ability to achieve high customer satisfaction	20		Y		X				
ability to meet specified and implied requirements	22						Y		
Staff productivity	22								
Staff morale/job satisfaction	21								

X indicates that a significant relationship exists between capability level and performance indicator
Y indicates that a significant relationship exists between attribute achievement and performance indicator

Some of the relationships are consistent with expectations, for example, quality assurance process and customer satisfaction performance; project management with the triple constraint of budget, schedule and meeting requirements. It was surprising that capability of the requirements elicitation process did not correlate with the ability to meet specified and implied requirements. The questionnaire also asked about the importance of these performance measures. All firms considered these measures to be important: the 22 firms perceived customer satisfaction and meeting requirements were important, and the majority highly rated the importance of the other performance measures (budget 82%; schedule 90%; productivity 92%; staff morale 87%).

In reports from the SPICE phase 2 trials, El Emam and Birk (2000a) refer to the relationship between the performance measures and the attribute achievement rating as the *predictive validity* of the process capability score. In evaluating predictive validity, they used only 4 of the 29 processes reported in the SPICE phase 2 trials. The four processes chosen were: develop software requirements (ENG.2), develop software design (ENG.3), implement software design (ENG.4), and integrate and test system (ENG.5). These four processes are included in the software development process in the RAPID method reported in this thesis. El Emam and Birk (2000a) found that for small organisations (less than 50 staff), schedule performance is associated with develop software design (ENG.3). For large organisations (more than 50 staff), multiple associations were reported: budget performance is associated with attribute achievement of develop software design (ENG.3) and implement software design (ENG.4); schedule, customer, requirements and staff morale performance are all associated with develop software design (ENG.3); and finally, productivity performance is associated with develop software requirements (ENG.2) and integrate and test system (ENG.5). In the PIP study, as there were only three firms with more than 50 FTE employees, this analysis was not conducted to compare the predictive validity of small and large firms.

8.3.4 Discussion focussing on processes at assessment

The previous section examined the organisation and project characteristics and how those factors related to the capability levels of the eight processes. This section will discuss each of the eight processes in terms of the assessment results.

To provide a richer context to the discussion, the assessment reports were analysed for common themes as 'placing exclusive reliance on a quantitative approach may result in limitations to the conclusions drawn from the assessment' (Rout, Neilson & Gasston 1994, p. 777). Themes identified included the involvement of the firm's managing director at the assessment, the need for tools for configuration management or problem tracking, shortage of staff, use of Intranet to facilitate communication, and reliance on key staff rather than procedures. A summary table showing themes identified for each firm is provided in Appendix J Table J.8.

Qualitative analysis of the advice given to the firms by the assessors was also undertaken. A summary of the assessors' advice is provided in Appendix J, Table J.9. The advice is grouped by three categories: standards and guidelines; tools and software; techniques, approaches and methodologies. For readability, in the discussion, the capability level is reported by the number, for example *level 0*, rather than by name: *incomplete*. The other capability levels are: level 1 *performed*; level 2 *managed*; and level 3 *established*.

As explained in §7.3.1, the capability levels in the process profiles of the 22 assessment reports were collated (the list of capability levels for the 22 firms is provided in Appendix I, Table I.6) and a summary is provided here as Table 8.10 showing the number of firms at each capability level for each of the eight processes.

Table 8.10 Distribution of firms at each capability level by process.

Process	Level 0: incomplete	Level 1: performed	Level 2: managed	Level 3: established
Requirements elicitation	1	10	9	2
Software development	1	13	8	0
Configuration management	3	13	4	2
Quality assurance	13	6	2	1
Problem resolution	5	14	1	2
Project management	4	11	6	1
Risk management	11	9	1	1
Process establishment	15	5	2	0

Requirements elicitation

The *requirements elicitation* process gathers, processes, and tracks evolving customer needs and requirements throughout the life of the software product. The requirements elicitation process exhibited higher capability compared to the other processes in almost all cases. Only one firm (#12) was rated at level 0; and 10 were assessed at level 1. Half of the organisations were rated at level 2 (9 firms) or level 3 (2 firms) for the requirements elicitation process.

Software development

The *software development* process is strongly linked to the requirements elicitation process, and aims to transform a set of requirements into a functional software product or software-based system that meets the customer's needs (ISO/IEC TR 15504-2 1998). Again, one firm (#13) had an incomplete process for software development, the majority achieved level 1 (13 firms) and 8 firms were rated at level 2. The assessment reports identified testing as a major issue of concern for 10 of the 22 firms (Appendix J Table J.8). This issue was highlighted recently in a report about the Australian software industry: developers rated insufficient time and resources to adequately test software as their second greatest concern behind lack of business opportunities (Sweeney Research 2003b; Woodhead 2003).

Configuration management

The configuration management process aims to establish and maintain the integrity of all the work products of a process or project (ISO/IEC TR 15504-2 1998). Although three firms were assessed at level 0, most were performing this process (13 firms), four were assessed at level 2, and two firms achieved level 3 for configuration management (#5 & #9).

Configuration management has been identified as a major issue for 60 percent of SMEs (James 2000), and the assessment reports highlighted the concern of firms in relation to selecting effective configuration management tools (Appendix J Table J.8). The assessors recommended tools such as Visual Source Safe to seven of the firms and only two firms were satisfied with their current configuration management tools (#4 & #9) (Appendix J Table J.9). In 1995, the SEI identified configuration management as a common weakness (Hayes & Zubrow 1995), but since then the availability and adoption of low cost tools has brought effective configuration management within the reach of even small firms (Dart 1992).

Quality assurance

The *quality assurance* process provides assurance that work products and processes comply with their specified requirements and adhere to their established plans (ISO/IEC TR 15504-2 1998). More than half of the firms (13) were assessed at level 0, six firms at level 1, two firms were assessed at level 2 (#4 & #18), and only one firm (#22) achieved level 3. As mentioned in §8.3.2, only two of the firms held certification to ISO 9001 (firms #8 & #22), although nine firms stated they intended to achieve certification (Appendix J Table J.8). The certified firms were rated at level 1 and level 3 for quality assurance. At the time of the assessment, the new version of ISO 9001 had come into effect and organisations needed to review the new standard prior to embarking on the certification process. Since 1996, when the Queensland Government abolished the mandatory policy for ISO 9001, the interest in quality assurance appears to have waned.

Problem resolution

The purpose of the *problem resolution* process is to ensure that all discovered problems are analysed and resolved and that trends are recognised (ISO/IEC TR 15504-2 1998). Five of the firms did not have a complete process in place for problem resolution, 14 firms were assessed at level 1, one at level 2, and two achieved level 3 (#4 & #9). The assessment reports revealed that 14 firms had insufficient measures to record and track problems (Appendix J Table J.8), and in eight cases, assessors recommended that the firm use a system to track problems (Appendix J Table J.9). To survive, small firms need to be customer focussed and the assessed firms saw this process as a high priority. Some firms had initially set up systems to track customer complaints and issues, then adapted these systems for internal use.

Project management

The *project management* process identifies, establishes, coordinates and monitors activities, tasks and resources necessary for a project to produce a product and/or service meeting the requirements (ISO/IEC TR 15504-2 1998). Four firms were rated at level 0 for project management, and these four firms, with a range of staff from 6 to

10 were not the smallest firms in the group assessed. Half of the firms were rated at level 1, six firms were assessed at level 2 and one firm (#7) achieved level 3. Although most firms undertook planning activities, 14 firms were reported as having insufficient measures to record development effort (Appendix J Table J.8). In eight cases, the assessors recommended the use of MS Project to improve the project management process (Appendix J Table J.9).

Risk management

Risk management is closely related to project management. The purpose of the risk management process is to identify and mitigate the project risks continuously throughout the lifecycle of a project. Half of the firms were assessed at level 0, 9 firms at level 1, one firm at level 2 (firm #7) and one firm at level 3 (firm #8). Reporting results of a SPICE-based survey of SME software development firms in Italy, Bucci, Campanai and Cignoni (2001) concluded that many SME developers have a naive approach as they identified risk management as a critical process but were unable to describe the process in terms of its goals or activities. A recent survey of 600 Australian small businesses conducted by the Certified Practising Accountants (CPA) in Australia (2002) found few small businesses have risk strategies in place, and where strategies are considered, they are unlikely to be formalised. The assessed firms with risk management processes were more concerned about commercial risk, rather than technical risk. Their market strategy was to understand the needs of the client and develop a close relationship to ensure the survival of the firm. This is consistent with the SEA report which identified the top priority for Australian software development firms was 'staying alive through the next 1-2 years' (Sweeney Research 2003b; Woodhead 2003).

Process establishment

The process establishment process is related to the quality management process in that it creates the processes against which the QA function checks work practices for conformance. Of the eight processes assessed, the process establishment process recorded the lowest capability with 15 firms rated as having an incomplete process, five at level 1, and only two at level 2. It may have been considered ambitious to include process establishment in the assessment of such small companies. Although only 7 of the 22 firms had a process for creating and enhancing process assets, this process was a valuable part of the RAPID assessment in that it raised awareness and provided the impetus to institutionalise the new processes which were spawned by the process improvement process. The assessment reports noted that in 13 of the firms, competent staff and informal standards were used rather than documented processes (Appendix J Table J.8).

Process interdependencies

The previous discussion focussed on each process individually; this section considers the relationships between the processes. As stated in §6.5, the primary processes of requirements elicitation and software development compose the core of the RAPID model. Higher capability in these core processes is supported by performance in management and organisation processes. The PIP manager was particularly interested in the capability of small firms in relation to the processes that support capability level 2 (managed): project management, quality assurance, configuration management, and problem resolution; and also the processes which provide a platform for improvement beyond the managed level to capability level 3 (established): risk management and process establishment (Rout et al. 2000).

To explore the links between the RAPID processes, the ISO 15504 assessment model and indicator guidance document (ISO/IEC TR 15504-5 1998) was analysed. As shown in Table 8.11, as well as links between processes, links exist between processes and the process attributes of other processes.

Table 8.11 Links from RAPID processes to other processes and process attributes

Process	Associated processes	Attributes supported by process
RE	The tracking of requirements is handled by CM	2.1 & 2.2
SD	The agreed requirements may be provided by RE	
CM	CM supports performance of the process attribute in those instances where it is invoked	2.2
QA	QA should be coordinated with and may make use of the results of other supporting processes such as PR	2.1 & 2.2
PR	PR supports performance of the process attribute in those instances where it is invoked	2.2 & 3.1
PM	For the identification of existing risks see RM. For the evaluation of risks see RM	2.1 & 3.1 & 3.2
RM	RM supports performance of the process attribute in those instances where it is invoked	2.1
PE	PE supports performance of the process attributes in those instances where it is invoked	3.1 & 3.2

(derived from Part 5 and annex b to Part 5 ISO/IEC TR 15504-5 1998)

Therefore, for a process to achieve level 2 (managed), it is expected to have at least capability level 1 for requirements elicitation, configuration management, quality assurance, problem resolution, project management and risk management, because level 2 (managed) attributes are supported by the performance of those processes. To achieve level 3 (established), it is suggested to have at least level 1 (performed) for project management and process establishment because level 3 attributes are supported by those two processes.

As shown in Appendix I, Table I.6, eight of the firms did not achieve higher capability levels than level 1 at the assessment. Four of the firms (#4, #7, #8, #22) conformed to the dependencies listed above. However, this was not the case for 10 of the 22 firms: the process attribute was achieved without the adequate performance of the corresponding supporting process. Table 8.12 lists the process capability levels for the 10 firms which exhibited inconsistent process levels.

Table 8.12 Firms rated at level 2 or 3 without achievement of supporting process attributes

Firm	Process capability levels								Inconsistency between process levels
Id#	RE	SD	CM	QA	PR	PM	RM	PE	
1	2	1	1	0	1	0	1	0	RE 2 but QA, PM 0
3	2	1	1	1	1	1	0	0	RE 2 but RM 0
9	1	2	3	0	3	2	0	0	SD, PM 2 but QA, RM 0 CM, PR 3 but QA, RM, PE 0
11	2	1	1	0	0	1	1	0	RE 2 but QA, PR 0
14	1	2	1	0	2	2	1	0	SD, PR, PM 2 but QA 0
16	2	1	0	0	1	1	1	1	RE 2 but CM, QA 0
18	3	2	1	2	1	2	0	1	SD, QA, PM 2 but RM 0 RE 3 but RM 0
19	1	1	2	0	1	1	1	0	CM 2 but QA 0
24	2	2	2	1	1	1	0	0	RE, SD, CM 2 but RM 0
25	2	1	1	1	0	1	0	2	RE, PE 2 but PR, RM 0

A reason for this inconsistency may be related to the scope of application of the process. For example, a firm achieving level 2 for requirements elicitation may have a satisfactory project management process in the early stages of the lifecycle, but project management may be inadequate in the later stages. The one-day limitation on the

RAPID assessment may also have contributed to this possible inconsistency, as could the inexperience of the assessors. Prior studies have not compared the achievement of the *managed* process attributes with the performance of the management processes, or the achievement of the *established* process attributes with the performance of the process establishment process.

8.3.5 Summary of discussion relating to research question 2

The analysis of the assessment reports confirmed that for the firms included in the process improvement program, the capability of technical processes is higher than that of management processes. Requirements elicitation and software development showed higher capability compared to the other processes. More than half the firms did not perform quality assurance or process establishment, and risk management was not performed at half the assessed firms. For some processes, higher capability is related to the proportion of experienced and highly qualified staff, the proportion of technical staff, firms with projects of lengthier duration, and certification to ISO 9001.

8.4 Discussion of findings related to research question 3

The third research question asked if assessment-based SPI programs are an effective means to improve process capability for small software development firms, and is the extent of improvement associated with particular organisational characteristics? A distinctive aspect of this study was that the firms provided with RAPID assessments were contacted for a follow-up meeting to ascertain the adoption of the recommendations from the assessment. Although many SPI assessments are performed and reported, with the exception of Goldenson and Herbsleb (1995), SPICE Project Team (1998) and Varkoi (2004), little research is available about the extent of improvement after an assessment. It is vital that assessments are followed up to measure the extent improvement, to find out why so many SPI attempts are abandoned, and to improve the SPI models and the methodologies.

In this section, the findings from the data analysis of the 20 follow-up meetings (presented in Chapter 7) are summarised and explained within the context of this and prior research which was examined in Chapter 2. Firstly, the extent of improvement determined at the follow-up meetings is discussed, then the association of organisational characteristics with the extent of improvement is explored. Particular issues which surfaced from the qualitative analysis of the final reports are presented, followed by lessons learned, then the impact of critical success factors identified in the theory are evaluated in relation to the PIP reported here.

Throughout this section, reference is made to the assessment and final reports, summaries of which are provided for each firm in Appendix H. Appendix I Table I.24 shows the capability levels and attribute achievement for firms by process as determined at the follow-up meetings by the assessors. A summary of themes identified from the qualitative analysis of the final reports is provided in Table J.10 in Appendix J.

8.4.1 Extent of improvement

Although follow-up meetings were held with 20 firms, only nine firms were formally reassessed. The quantitative analysis of the extent of improvement of capability levels and attribute achievement measures involves only the nine formally reassessed firms.

A comprehensive summary of the results from the RAPID program was presented in Chapter 7, Table 7.39, and to facilitate discussion, the 22 firms are grouped according to the success of the outcomes of the program as summarised in Table 8.13.

All the firms in groups 1 to 3 were formally reassessed:

- Group 1: This group comprises six firms which increased the capability level of at least one of the eight processes (#9, #8, #18, #5, #13, #21)
- Group 2: One firm showing improvement to attribute achievement, but not enough to increase the capability level rating of any process (#23)
- Group 3: In this group of two firms, improvements were reported to some of the eight processes, but not enough to increase the capability level or attribute achievement rating of any process (#14, #16).

The firms in groups 4 and 5 were not formally reassessed, but their progress was recorded by the follow-up assessor by way of an on-site follow-up meeting, or telephone call:

- Group 4: These six firms reported limited improvements (#7, #25, #3, #19, #1, #17)
- Group 5: Completed program: five firms reported no improvements as a result of the assessment (#4, #22, #11, #15, #12).

Follow-up meetings were not held for the two firms in group 6 as firm #24 withdrew from the program, and firm #2 could not be contacted.

As shown in Table 8.13, formal reassessments were conducted in nine of the firms, and with the limitation of half a day for the follow-up meeting, such reassessments focussed on improvements since the initial assessment. Therefore, it is possible that some of the processes in place at the firms had actually suffered a decrease in capability. It is acknowledged that the formal reassessment focussed only on the eight processes covered at the initial assessment, and furthermore, on the recommendations provided at the initial assessment. The temporal and financial restrictions of the process improvement program prevented the opportunity for a thorough reassessment.

Table 8.13 Groups of firms based on outcome of PIP

Reassessment	Formal			Informal		None
Group #	1	2	3	4	5	6
Extent of improvement	Capability level improved	Attribute achievement improved	Specific processes improved	Limited improvement	No improvement	Withdrew from program
Firm #	#5, #8, #9, #13, #18, #21	#23	#14, #16	#1, #3, #7, #17, #19, #25	#4, #22, #11, #15, #12	#2, #24

The comparison of capability before and after the process improvement program only included the 9 firms that were formally reassessed. As detailed in §7.3.11, there was significant improvement in capability levels for four of the eight processes: software development; configuration management; project management; and process establishment. Also, attribute achievement improved significantly for three processes: configuration management, project management and process establishment (statistical tests in Table 7.41).

Therefore, to answer research question 3, the process improvement program, based on the RAPID methodology was effective for these small firms: not only did 15 of the 20 provide evidence of improvement, but the improvement in four processes of the nine reassessed firms was significant (§7.4.1). This positive finding of the effectiveness of the SPI program is in contrast to that of Richardson (1999) who concluded that current software process models were not suitable for small Irish software development firms because they were 'cumbersome, costly, failed to present a comprehensive improvement strategy and did not show the effect of practice improvement on all processes' (p. 217).

In the next section, selected organisational characteristics and their association with the extent of improvement is discussed (§8.4.2), followed by a more detailed discussion of the formally reassessed firms (groups 1, 2 and 3) in §8.4.3, then the informally reassessed firms (groups 4, 5 and 6) in §8.4.4.

8.4.2 Organisational factors

In Chapter 2, previous research (Brodman & Johnson 1994; Deephouse et al. 1995; Goldenson & Herbsleb 1995) suggested that some organisational factors may facilitate the adoption of the recommendations thereby resulting in greater extent of improvement. This research had varied results about these factors. As reported in §7.4, the hypotheses suggesting that extent of improvement would be related to firm size (§7.4.4) and the proportion of experienced staff (§7.4.5) were not supported. Formal educational qualifications are associated with the extent of improvement in configuration management and process establishment (§7.4.6). Furthermore, the proportion of technical staff is associated with the extent of improvement in software development and process establishment processes (§7.4.7).

Although organisational factors such as size have been discussed in many reports, Beecham, Hall and Rainer (2001) concluded from their study of SPI problems in 12 UK companies that organisational issues are more of a problem for high maturity firms, and that project and technical problems are the concern of low maturity firms. Therefore, qualitative analysis of final reports was undertaken to identify factors which enabled or prevented successful SPI by the firms in this study.

8.4.3 Formal reassessments

A stated earlier, nine firms were formally reassessed during the follow-up meetings. In this section, the improvement in capability levels is discussed, followed by the improvement in attribute achievement.

Capability level improvement

The six firms in Group 1 increased the capability level of at least one process as shown in Figure 8-3 (full list in Appendix J, Table J.11). The extent of improvement varied from the most improved firm (#8) with seven of the eight processes improved, to firm #21 which improved one process.

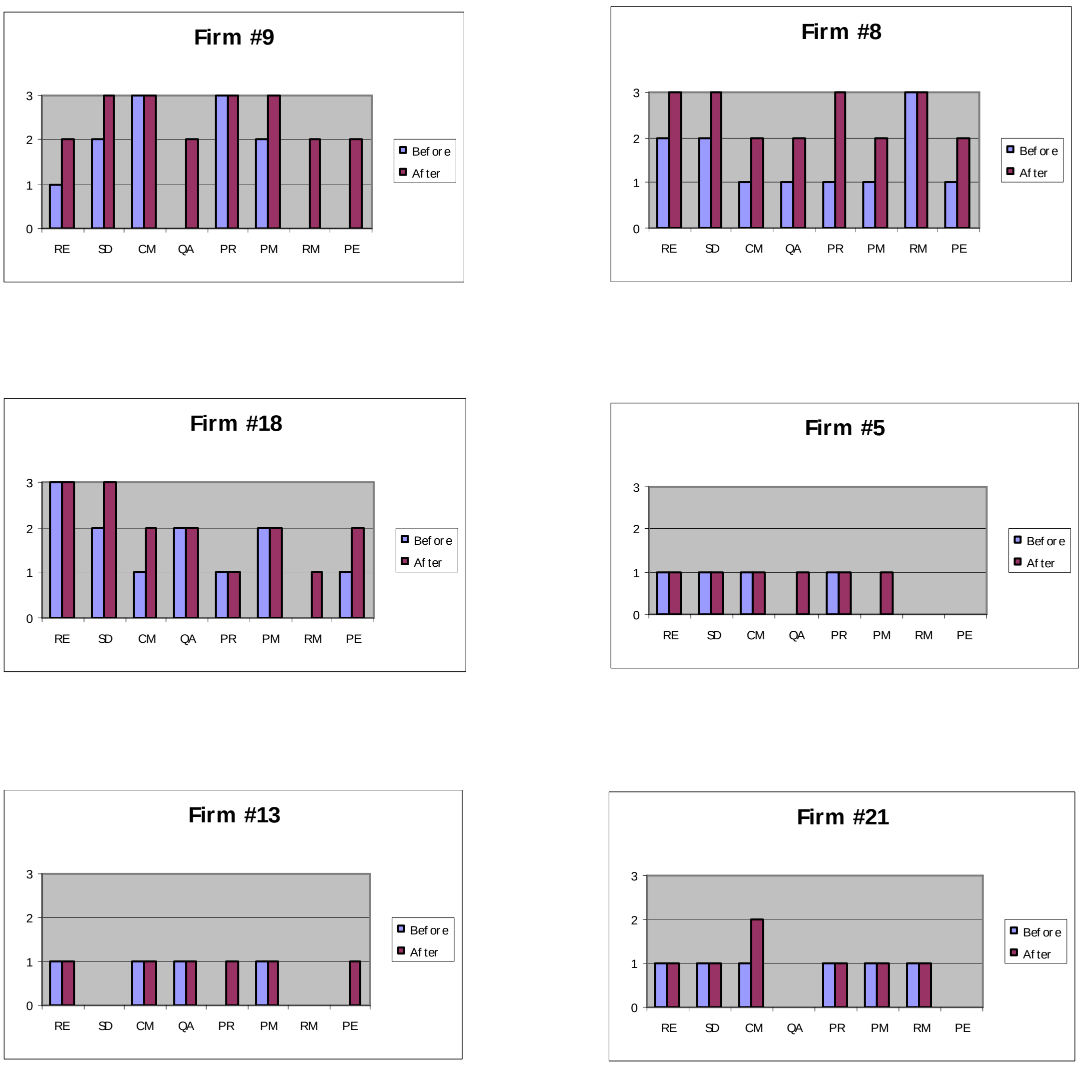


Figure 8-3 Process capability levels at initial assessment and follow-up meeting for group 1 firms

Analysis of the process improvement of the group 1 firms showed that the process establishment process, with the lowest capability levels at the initial assessments, showed the greatest extent of improvement. At the other extreme, requirements elicitation, which had scored the highest capability levels at the initial assessments, showed the least improvement. This suggests that many firms found process establishment relatively easy to improve. Based on standards suggested by the assessors, current informal procedures to create process assets were documented and formalised.

A distinct and valuable element of the RAPID method was the inclusion of process establishment as any new processes spawned from and since the program will be implemented, ensuring that the program will have a more lasting effect.

Attribute achievement improvement

To achieve higher capability levels, all the firms in group 1 improved their attribute achievement. In addition, there was one firm, assigned to group 2, which had improved in terms of attribute achievement, but not enough to change its capability level. Firm #23 improved the performance management attribute (PA2.1) of the project management process from partially to largely achieved. This was the only case of attribute achievement of a firm which did not improve its capability level. This improvement left the process capability level unchanged at level 1.

Figure 8-4 displays the improvement in attribute achievement for the seven relevant firms. As explained in §7.3, the assessment rated the extent of achievement of each of the five process attributes as not achieved, partially achieved, largely achieved or fully achieved. This rating was converted to an ordinal scale: not achieved—0; partially achieved—1; largely achieved—2; fully achieved—3. Therefore, in Figure 8-4, the Y axis has a range from 0 (all five attributes not achieved) to 15 (all five attributes fully achieved).

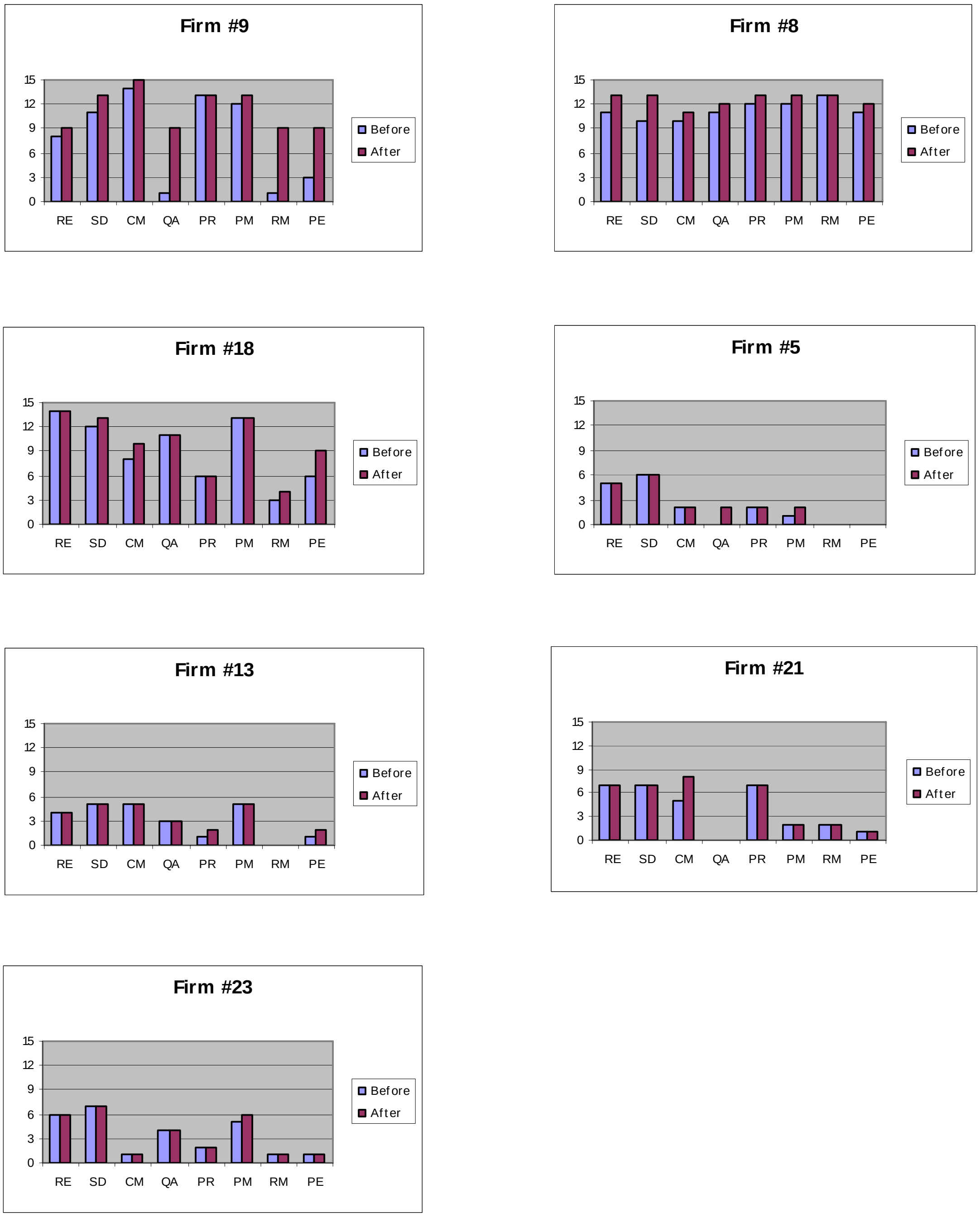


Figure 8-4 Attribute achievement at initial assessments and follow-up meetings group 1 and 2 firms

Process improvement

Two firms were assigned to group 3. Firm #14 and #16 were formally reassessed but did not achieve increases in capability levels or attribute achievement. The final report for firm #14 included comments about improving the process performance attribute (PA1.1) of quality assurance and process establishment (both were initially assessed at partially); and firm #16's final report included positive comments related to process performance attribute (PA1.1) for quality assurance and configuration management (both initially assessed at partially), but none of these improvements were deemed to be sufficient to change the process attribute achievement from partially to largely achieved.

8.4.4 Informal follow-up meetings

The 11 remaining firms which completed the PIP did not have a formal reassessment. There were two main reasons why a formal reassessment was not conducted: either it was established by the assessors that the firm had not implemented enough of the recommendations to be formally reassessed, or the firm declined the offer of the site visit and discussed their SPI progress with an assessor by phone.

Six of these firms had implemented some of the recommendations from the assessment and on that basis were assigned to group 4. The improvements recorded by the follow-up assessor were not specific to the eight processes, and included the development of templates; evaluation and implementation of tools; review of business goals; formalisation of testing procedures; and establishment of measures such as actual effort. Some of the firms in group 4 reported that their efforts had been stymied by such issues as: high staff turnover including loss of key staff (#17); relocation of business (#7); burglary of premises (#19); changed business focus (#1, #7); and disruption due to GST work (#17).

Five firms completed the program but did not report any process improvement. These firms were assigned to Group 5. Four of these follow-up meetings were conducted by phone (#12, #15, #22, #40). When asked why the recommendations had not been implemented, a wide variety of reasons was provided:

- restructuring of business (#11)
- personal health problems (#4)
- workloads due to GST and Y2K demands (#12)
- lack of SPI expertise, needed a mentor (#15)
- did not implement recommendations as initial assessment was of minimal value (#22).

Firm #2 could not be contacted as it had ceased to operate prior to the follow-up meeting. It was not surprising that some of the 22 firms had changed the focus of their business, or ceased to operate over the time period from the assessments to the follow-up meetings. In their study of small Italian software firms, Raffa, Zollo and Caponi (1996) found that most of the firms they surveyed operated as software developers for three to seven years, and were forced to significantly reduce their involvement in software development, shifting their strategic focus to the commercialisation of hardware and software, and provision of other information services. A study which analysed Australian business changes in ownership and cessations noted that exit rates are higher for smaller businesses, regardless of the age of the business, with around 35 percent of small businesses exiting in their first five years of operation (Bickerdyke, Lattimore & Madge 2000).

8.4.5 Summary of discussion relating to research question 3

Based on the findings above, it is clear that assessment-based SPI programs are an effective means to improve process capability for small software development firms, but the extent of improvement, although associated with the proportion of qualified and technical staff, appears to depend on a more complex set of conditions. These critical success factors, relating to the firms and the RAPID method, will be explored in the next section.

8.5 Critical success factors for SPI in small firms

The qualitative analysis of the assessment and final reports identified many issues related to SPI for small firms. In this section, insights gained specifically related to the firms participating in the study are explored (§8.5.1), followed by a discussion of the issues relating to the plan and execution of the PIP (§8.5.2). The next part (§8.5.3) considers the relevance of other critical success factors (CSFs) that were identified in the literature.

8.5.1 Specific issues relating to the PIP firms

Detailed qualitative analysis of the 22 assessment reports and 20 final reports prompted further investigation of specific issues related to the 22 firms. This section discusses the elapsed time from the assessment to follow-up meeting, the need for mentoring, the readiness of firms for SPI, the role of firm owner/manager, and finally the advice provided by the assessors.

Elapsed time from assessment to follow-up

In formulating the process improvement program, the PIP manager planned for the follow-up meetings to be conducted 6 months after the initial assessment to evaluate the effectiveness of the program. When the firms were contacted by the assessors for the follow-up meeting, many wished to defer the meeting due to current workloads. In fact, the time period from the initial assessment to the follow-up meeting ranged from 7 to 16 months, with a mean of 12 months and standard deviation of 2.4 months.

As shown earlier in Table 8.13, the follow-up meetings conducted were either formal reassessments of the capability of some or all of the processes, or an informal follow-up meeting discussing the extent of adoption of the recommendations. To evaluate the relationship between the type of follow-up meeting and the time period from the initial assessment to the follow-up meeting, an analysis of variance was performed. As shown in Appendix J, Tables J.12 and J.13, there is a significant difference: firms which were formally reassessed held their follow-up meetings after a shorter time period than firms not formally reassessed. The extent of elapsed time could have been influenced by how promptly the assigned assessor contacted the firm to arrange the follow-up meeting, but in many cases, firms deferred the follow-up meetings, citing work commitments and pressing deadlines.

To further explore the relationship of time period and program outcome, Spearman rank correlation tests were performed for the process capability levels of the nine formally reassessed firms. The statistical analysis indicates that longer time periods (from assessment to follow-up) are associated with lower process improvement for quality assurance and project management (test results in Table 8.14).

Table 8.14 Association: extent of process improvement and number of months from assessment to follow-up meeting

Difference in capability levels	RE	SD	CM	QA	PR	PM	RM	PE
r_s	-.313	-.552	-.230	-.608(*)	-.127	-.644(*)	-.449	-.470
p (1-tailed) (n=9)	.206	.064	.276	.042	.373	.031	.113	.101
Difference in attribute achievement	RE	SD	CM	QA	PR	PM	RM	PE
r_s	-.311	-.509	-.250	-.629(*)	-.104	-.568	-.449	-.509
p (1-tailed) (n=9)	.206	.082	.259	.035	.395	.056	.113	.081

The finding that a shorter follow-up period was more effective in the PIP study is in contrast to the conclusion reached by Varkoi (2004) in Finland. After analysing results from a SPICE-based SPI program involving 20 small firms in Finland, Varkoi (2004) decided to extend the time frame from 6 months for the pilot phase to 12 months for the harvesting phase, although the participants in the study considered two years to be the optimal length for an improvement program.

The assessment report provided recommendations to the firms based on a six month time-frame. This is consistent with the view held by Debou and Kuntzmann-Combelles (1996) who urge that a 3-5 month timeframe for action plans be considered, and that it is better to adopt a narrow focus of improvement actions. The problem with a six month time-frame is that many firms (such as #3, #8, #16, #23) had designed new processes, but had not yet used them at the time of the follow-up meeting. This confirms the view of Paulk et al. (1995) and Krasner (1997): it can take two years for process changes to demonstrate results.

It appears that more research is needed to investigate the optimal time period from assessment to reassessment. Although planning needs to encourage achievement of short term goals, many rewards are not evident until a much longer timeframe.

Mentoring

Small firms need external assistance in planning and implementation of the improvements as they have scarce resources and limited possibilities to keep up-to-date with the state-of-the-art research and practice (Kautz 1998a; Varkoi 2004).

On-going mentoring was not provided to firms although three firms mentioned that lack of mentoring inhibited their SPI progress (#7, #8, #15). At the same time as the PIP project was conducted, two staff from SQI were providing ongoing mentoring for two firms through a complementary program, *SEA Showcase.* Two of the PIP firms expressed disappointment that they were not selected for the Showcase program.

To facilitate the necessary technology transfer for SPI, the role of mentor may be critical to the success of SPI programs. The effectiveness of mentors in SPI programs has been documented with the role of mentors promoted to include 'motivating, advising, supporting, encouraging, teaching, listening, solving problems, calming fears, and assisting in artefact collection' (Reeb & Henderson 1997, p. 6). Herbsleb and Goldenson (1996) analysed 138 survey responses from CMM assessed organisations and found that three quarters of these organisations understood what needed to be improved, but needed more guidance about how to improve, and more than half needed more individualised mentoring and assistance (p. 328). An analysis of 37 high maturity organisations revealed that half of these successful organisations have a 'formal mentoring program to impart skills and knowledge' (Paulk, Goldenson & White 2000, p. 8).

The analysis of the process improvement program reported here supports the view put forth by Thong, Yap and Raman (1996, p. 248): for small businesses operating in an environment of resource poverty, high quality external expertise is even more critical than top management support.

Readiness for SPI

After analysing reports which indicate that the vast majority of organisations in the US and UK are at the initial level of maturity, Smith et al. (1994) assert that it is clear that only a handful of companies are ready for SPI 'because their software health is so bad (that is if they have any development process at all)' (p. 207). They go on to warn that in order to be ready for SPI, a visible and defined software process must already be in place. The opinion that low maturity organisations find it much more difficult to change and implement SPI is shared by Diaz and Sligo (1997) based on these reasons: low maturity firms do not collect metrics; they focus on defining core processes, not on improvement; and it takes a lot of effort to get started to overcome scepticism and to be sure of management support and long term commitment. Other researchers also believe it is pointless to try to implement high maturity processes into low maturity projects (Hall, Rainer & Baddoo 2002; Kaltio & Kinnula 2000). Recently, Rainer and Hall (2002) determined that factors impacting on SPI adoption varied for low maturity and high maturity organisations.

To determine if higher initial capability levels are associated with the extent of improvement, correlations were calculated (for the nine reassessed firms) for each of the eight processes. Results show that only for software development is such an association evident (see Appendix J, Table J.14), indicating that firms with higher initial capability for software development showed higher improvement in that process.

It is interesting to consider the performance of the five firms (#4, #7, #8, #18, #22) which, at the time of the initial assessment, were rated level 1 or higher for at least seven of the eight processes. In this discussion, these five firms are classed as high capability and the remaining 17 firms are referred to as low capability firms. As shown in Appendix J Table J.11, two of the high capability firms (#8, #18) are included in Group 1, having achieved sufficient improvement to increase the capability level of some of the eight processes. Two of the other highly rated firms (#4, #7) experienced seriously disruptive events which they reported prevented them from implementing the recommendations from the assessment. The remaining high level firm, #22 expressed the opinion that the RAPID assessment was too brief to be of any value.

However, some of the firms with low initial capability were also successful in the program. The gains achieved by the four low capability firms (#5, #9, #13, #21) in Group1 were certainly more modest than those of the higher capability group, but still a notable achievement. Furthermore, seven low capability firms (#16, #23, #25, #3, #19, #1, #17) reported that they had successfully implemented some of the recommendations, citing improvements in terms of defining their methodologies, developing templates, recording problem reports, and formalising testing procedures.

Therefore, this research indicates that low-rigour SPICE-based assessments are effective for small firms with poorly defined processes.

Role of firm owner/manager

An interesting aspect of the PIP was the high involvement by the owner of the firm. In 14 of the 22 firms assessed, the 'managing director' or 'company director' was explicitly recorded in the assessment report as attending the assessment (Appendix J, Table J.8). This owner/manager role is a characteristic of small business, for example, 70 percent of Australian small business operators were classified as full-time operators (ABS 2004a). However, the program outcomes in this study did not vary significantly depending on whether the managing director was present or not (Appendix J, Table J.15 has a list by outcome group showing number of firms with managing director involved).

Lack of senior management commitment is recognised by Debou and Kuntzmann-Combelles (2000), Abrahamsson (2001), El Emam et al. (1998), and Wilson, Hall and Baddoo (2001) as a major bottleneck to the success of SPI initiatives, but for most small firms, the business operator is often involved in all aspects of the business and would therefore instigate the SPI and participate heavily in it.

Advice provided by assessors

All assessors had completed the SPICE certification training course, ensuring the consistency of capability levels ratings. However, the various assessors provided a valuable and diverse range of advice to the firms, drawing on their personal knowledge and expertise. The advice provided by the assessors is summarised in Table J.9 in Appendix J and has been grouped into three categories: standards; tools; and techniques/approaches/methodologies. The specific advice provided most frequently was MS Project (8 instances), Visual Source Safe for configuration management (7 instances), and the Project Management Body of Knowledge (PMBOK) (5 instances).

8.5.2 Meta analysis of PIP: lessons learnt

In this section, issues relating to the PIP method are raised, and improvements to the PIP procedures are suggested. In Chapter 9, specific recommendations are made to practitioners and consultants.

Comments from the firm sponsors were gathered through feedback questionnaires and by the follow-up assessors. Most of the firms enthusiastically commended the process improvement program, commenting that it was an effective introduction to SPI; that it provided an accurate review of the current status of development processes; and that it motivated them to improve their planning and documentation. Many expressed regret that they were unable to put more resources into implementing the recommendations, but the timing of the program clashed with two urgent deadlines: the modifications for year 2000, and the introduction in June 2000 of the Australian Government's Goods and Services Tax legislation.

Two of the firms (#7, #8) had hoped to be included in the more intensive *Showcase* program, and lost motivation when funding for that program was reduced. Negative comments were made by only one firm (#22). Firm #22 was the largest included in the program, and felt that a one-day assessment was too brief to be of any real value.

Role of assessor

The PIP manager intended for the follow-up assessment to be conducted by one of the assessors who performed the initial assessment, but due to limited SQI staff availability, this was not always possible. Examination of Table 8.15 (based on Table 6.6) reveals that in three of the Group 5 firms, the follow-up assessor was not one of the initial assessors. If one of the initial assessors had contacted the firm for the follow-up, then the follow-up may have been more effective in terms of providing feedback about improvement progress or lack thereof. The people at the firm had formed a relationship with the two initial assessors, and a level of trust may have been established. To introduce someone new at the time of the follow-up meeting may have caused some anxiety for the firm sponsor, and the staff at the firm may have felt that the new assessor would not understand how the firm operates. They may resent the need to explain everything again, and also may also be worried about confidentiality. (A full list of initial and follow-up assessors is included in Appendix J Table J.16.)

Table 8.15 Summary of effect of consistent follow-up assessor for each outcome group

Group	Outcome	Firms in group	Firms where an initial assessor conducted the follow-up meeting
Group 1	Improved capability	6	5
Group 2	Improved attributes	1	1
Group 3	Improved processes	2	1
Group 4	Limited improvement	6	4
Group 5	No Improvement	5	2

Research has shown that 'small firms are averse to consultants and reluctant to seek external help' (Cragg 2002, p. 277). This was confirmed by Hall, Rainer and Baddoo (2002) who found that companies did not highly value the input of external consultants. Therefore, the assessors, as external consultants, need to develop a relationship with the developers in small firms. One of the lessons learnt in the SataSPIN project (Varkoi 2004) was the need for continuous contact, as well as contacting the firms at least once per month. Varkoi recommended that assessors also make contact with more than one person at each firm.

Cost benefit analysis

Only one of the follow-up meetings recorded an estimate of the investment made by the firm. Firm #17 reported that the program consumed 155 hours of staff time and included the purchase of Visio software. Most of the firms did not know the extent of resources involved because they did not have a measurement process in place. Low maturity firms typically do not have metrics for effort or defects. Each firm invested time in preparation and involvement in the RAPID assessment and follow-up meetings. At each firm, senior members of the development teams worked with the assessment sponsor to review the recommendations and formulate action plans. The effort of each firm in implementing the actions varied. Some firms released staff to attend training courses or to evaluate software development tools; others incurred costs to purchase and implement tools.

As evident from the follow-up meetings, the main benefits included improved quality assurance, configuration management, project management and testing. All companies improved the standard of their documentation, a move which has already returned dividends for one company which lost a key developer. A further important benefit to one company was the competitive advantage provided by quoting the capability ratings in promotional material.

The PIP could be improved by including a procedure for the follow-up meeting in the RAPID method. Although a template for the final report was provided to assessors, limited guidance was provided. It is also recommended that firms are requested to keep a record of SPI effort, costs and benefits. Such a record could be summarised in the final report, and published as success stories of SPI for small firms. These accounts of SPI success would encourage other small firms to embark upon process improvement. Managers are loath to adopt standards without information about trade-offs between increase in quality and cost of achieving that quality (Pfleeger, Fenton & Page 1994, p. 78).

Communication

Communication has been identified as a key factor in implementing SPI (Goldenson & Herbsleb 1995; Kautz, Hansen & Thaysen 2001; Wilson, Hall & Baddoo 2001). Small firms have an advantage in this regard. It is easier than in a large organisation to explain events and phenomena associated with program, to clearly define and document roles, resources, and responsibilities, and to clearly state the process goals. It appeared that in most of the firms, the sponsor effectively communicated the recommendations and action plan to the development team.

8.5.3 People, organisation and implementation issues

As discussed in §2.3.4, there has been a concerted research effort to define the critical success factors for SPI, and an extensive range of factors has been suggested as enablers and inhibitors of successful SPI programs. However, most of the research to date has reported on adoption of SPI by large organisations; there has been very little about SMEs. Some of the relevant factors have already been discussed (timeframe from assessment to evaluation, mentoring, readiness, roles of owner/manager and assessors). In this section, the impact of other factors is discussed in terms of their relevance to the small firms involved in the PIP, and their effect of enabling or inhibiting the PIP. Evaluating the process improvement program and its associated outcomes against these factors provides the opportunity to validate the importance of these factors for small firms, and to derive a set of critical success factors relevant to small software development firms.

People issues

The roles of the owner manager and assessor have been discussed earlier, but there are other people issues to consider. Reporting a study from Slovenia, Horvat, Rozman and Gyorkos (2000) found the influence of human factors was more important in small rather than large companies 'because of the important role of each individual in the small company' (p.52).

In large firms, organisational politics is identified as a key barrier to SPI success (El Emam et al. 1998; Goldenson & Herbsleb 1995). In the PIP, organisational politics may have been defused because the SPICE assessors, as external change agents with authority from the sponsor, were seen as removed from the internal company politics and were outside the scope of internal 'turf wars'.

Also, it is critical that the people involved in performing the assessments are respected (Goldenson & Herbsleb 1995; Wilson, Hall & Baddoo 2001). This was achieved in the PIP as all the RAPID assessors had completed the SPICE certification training and most were experienced assessors. Also, their credibility was enhanced by the reputation of SQI. SQI provides a focus in Queensland for expertise in software quality and serves as a catalyst for innovations in software quality techniques. It is engaged in a program of research with the local software industry and provides consulting and professional support to industry on setting up and managing software quality systems and on using national and international software standards (SQI 2001).

In a large organisation, it may be impossible to involve the company technical staff in the assessment (Goldenson & Herbsleb 1995), but in a small firm this is not a problem. Training and expertise is important (Debou & Kuntzmann-Combelles 2000; Kaltio & Kinnula 2000; Rainer & Hall 2001) but although appropriate training courses and seminars were conducted by SEA (Qld), many of the firms were unable to participate due to heavy workloads. In comparing best practice across firms from many and varied industries and countries, Voss et al. (1998) found that small firms (from 20 to 100 staff) neglect training compared to larger firms. As reported in §8.2.1, training for project managers and software engineering technology scored poorly in the best practice survey in this study: only 21 percent of organisations have adopted those practices.

Organisation factors

Recently, high costs and inadequate resourcing have been found to be one of the greatest impediments to SPI success (Bucci, Campanai & Cignoni 2001; El Emam et al. 1998; Hall, Rainer & Baddoo 2002). It seems a common problem that although companies have objectives for SPI, they do not adequately resource them (Goldenson & Herbsleb 1995; Hall, Rainer & Baddoo 2002), and this issue has been mentioned in the final reports as a key factor in limiting the implementation of the recommendations of the PIP by these small firms. Most of the organisations, especially the smaller ones, would not be prepared to pay commercial rates for such an assessment and many expressed appreciation to SEA for the opportunity to participate in such a valuable program. The resourcing problem extends beyond the cost of the assessment; despite the program being offered at no cost, many firms were unable to make resources available to fully implement the recommendations.

Organisational stability proved problematic for some of the PIP participants. Three of the firms suffered from instability during the program due to changes to their business focus, and major restructuring. It is not feasible to conduct an effective SPI program when the business goals are in a state of flux (Rainer & Hall 2001).

Implementation

Infrastructure encompasses human as well as IT resources. Unlike large organisations, small firms do not have SPI steering groups, Software Engineering Process Groups (SEPG) or control boards made up of representatives from different projects and functions to research and promote SPI and to 'provide points of coordination on crucial issues' (Hall, Rainer & Baddoo 2002 p. 5). However, the firms in this study were able to join the SEA SPI Special Interest Group and this provided opportunities to network with other firms engaged in SPI and to attend local presentations by SPI experts.

As far as technology infrastructure, Kaltio and Kinnula (2000) report the successful implementation and use of an intranet system for the process library at Nokia. In the PIP study, five firms mentioned they were developing an intranet to enable process assets to be shared by developers (Appendix J Table J.8).

SPI should be aligned with organisational goals (Rifkin 2001) and tailored to the individual needs of each organisation (Kautz, Hansen & Thaysen 2001; Varkoi 2004). Although the PIP assessors were mindful of the need to build on existing practices, the program did not allow for customisation in terms of the selection of processes. Warnings were also issued (Debou & Kuntzmann-Combelles 2000) against attempting to cover too wide a scope in the action plan. As the RAPID assessment only covered eight processes, the scope was defined to a manageable extent, but for some firms, the recommendations were not feasible in the six month time frame.

The primary success factor identified by Goldenson and Herbsleb (1995) and endorsed by Hall, Rainer and Baddoo (2002) was that managers actively monitor the progress of the process improvement program. At the initial assessment, each sponsor agreed that the outcome of the RAPID assessment would be evaluated by a follow-up meeting. This commitment by the sponsor (the owner or a senior executive in each firm) ensured progress was monitored so that, as far as possible, agreed recommendations were implemented prior to the follow-up meeting.

8.5.4 Summary of critical success factors

Based on the same framework used to structure the literature review in Chapter 2, the critical success factors are summarised in Table 8.16. The analysis revealed that some factors are critically important for small firms:

- Process improvement program must be low cost, or sponsored by government
- Short time frame to evaluation (but do not expect early benefits)
- Restricted scope, mini assessment
- Need external help from assessors/consultants
- Mentoring to develop ongoing relationship of firm with assessors

- The firm needs to have organisational stability; it is not advisable to attempt SPI during restructuring or other major organisational change
- Small firms do not take advantage of training courses, therefore need to involve mentors to provide on the job training.

There are some factors that have attracted much attention by researchers in respect to SPI adoption by large firms, but do not appear to be issues for small firms:

- Senior management commitment
- Organisation politics
- Communication within the development group.

To summarise the analysis, of the 22 firms which participated in the process improvement program, six firms improved process capability, and a further nine firms reported other improvements. All eight processes in the RAPID model showed capability level improvement, but most improvement was from process establishment, and least from requirements elicitation. The study demonstrates that small firms with poorly defined processes can benefit from SPI. Six months is suggested as an effective time frame for achieving objectives and undertaking an evaluation, as a longer evaluation period was not associated with increased improvement.

A distinct and valuable element of the RAPID method was the inclusion of process establishment as any new processes spawned from and since the program are implemented, ensuring that the program will have a more lasting effect. This study highlights the importance of the role of the assessor, although it adds to the cost of the program, small firms need ongoing contact with assessors as mentors.

The next section compares the survey findings with the findings from the process improvement program.

Table 8.16 Summary of CSF's impact on PIP

CSF	Item	PIP impact and comments
Economic factors	ROI, payback period	6 months was good for follow-up, but not long enough to see implementation. Should have collected cost/benefit data.
People issues	Management commitment	In small firms, managers are involved in all aspects, commitment is guaranteed.
	Staff involvement	Need to have the same assessor for assessment and follow-up meeting. Assessors should contact each firm once per month. Assessors accepted as 'SEA/SQI' experts. Some firms said it was difficult to unfreeze the current processes.
	Mentors	Small firms need (external) mentors.
	Politics	Not a problem in small firms.
	Training	Many did not take advantage of appropriate, local training.
	Motivation	+ve: follow-up meetings. -ve: no cost—firms had not invested in PIP except for time for assessment.
Organisation factors	Communication	Communication within firms not problematic. Request the sponsor disseminate information about the assessment model and method to staff prior to the initial assessment.
	Resources	Implementation of recommendations restricted by lack of resources.
	Business Strategy	Some firms changed their business strategy and this impacted (instability). Many said the assessment provided a valuable opportunity to evaluate their business goals.
Implementation	Infrastructure	Firms developed Intranet for process definitions, templates. SEA SPI Special Interest Group provided professional network.
	Realistic objectives	Some of the recommendations were not realistic within the 6 month timeframe.
	Tailor SPI	Method tailored from ISO 15504 to RAPID, but not specifically to each firm.
	Evaluation	Evaluation planned at time of assessment – provided motivation. Suggest use of a feedback form after follow-up meeting. Evaluations could be improved by developing better relationship with assessor
	Readiness	Assessment based-SPI is effective for low maturity firms.

8.6 Comparison of survey and process improvement program

As stated in §3.5.1, the research approach incorporated the use of the best practice survey as well as in-depth analysis of the PIP participant firms. The survey provided a broad view of the industry as a whole, and the analysis of the PIP reports gave a more detailed picture of a few companies, and also richer data. As well as differing in the range of subjects, the study used multiple data collection strategies. The survey was self administered, in contrast to the PIP field experiments where the assessment data was collected by structured interviews. Furthermore, as well as the quantitative analysis undertaken to statistically test the hypotheses, extensive qualitative analysis was undertaken of the assessment and final reports.

As the ESI survey was partially based on the SPICE standard and the RAPID method was wholly based on SPICE, it was expected that the survey results regarding best practice adoption would correlate with the RAPID results of capability. In §7.5, the survey adoption levels and PIP capability levels are compared, but the correlation between the survey and the PIP assessment is not significant. Although all assessed firms were members of SEA (Qld), some had not responded to the survey. Hence, matching survey and RAPID data were available for only 16 of the 22 assessed firms. The lack of correlation may stem from the fact that the capability level is derived from five process attributes: process performance; performance management; work product management; process definition; and process resource, whereas the survey only records the existence of a process. However, even when the analysis focussed on the comparison of the process performance attribute from the assessments with the corresponding processes from the survey, no correlations were found.

Other researchers have also found that the use of multiple methods often results in little convergent validity (McGrath, Martin & Kulka 1982). The comparison of the results from these two empirical evaluation projects raises some critical issues related to data collection. It is recognised that in using self-administered surveys such as the best practice questionnaire, the researcher forfeits the opportunity to verify that an appropriate person is completing the survey, that the questions are correctly interpreted and that the responses are honest. The ESI (1997) warns that results from a survey such as this provide a rosy view of the industry and that best practice could be exaggerated due to optimistic self-reporting. The ESI questionnaire is designed to assess the existence, not the adequacy, of best practices (Dutta, Lee & Van Wassenhove 1999). On the other hand, the RAPID assessment evaluates the existence, use and further development of software development processes. This study confirms the view held by Dutta, Lee and Van Wassenhove (1998) that the actual capability of the ESI survey respondents for the different software practices is lower than what is reported in the survey.

But even though the results did not correlate by process, from the summaries of survey findings (§8.2.9), assessment findings (§8.3.5), and follow-up findings (§8.4.5), some common themes emerged.

Both studies found that a wide variation exists in the level of adoption of best practice, with practices of a technical nature more widely adopted compared to techniques related to support and management. Also, within both sets of respondents, there was wide variation between the capability and adoption of the leaders compared to the laggards, indicating a large difference in the work practices of Queensland development firms. Furthermore, the survey showed that many organisations are interested in SPI, and the PIP proved that small firms can benefit from SPI, even when they commence SPI from a state of low maturity. Both studies highlighted similar weaknesses in terms of staff training, and the collection and use of actual effort for project management.

8.7 Summary

The discussion presented here highlights the wide variation in the extent of adoption of software development best practice techniques by software developers in Queensland. It was established that COTS developers have higher adoption than firms which do not develop COTS systems, and adoption is associated with the size of the development group. Organisations from the finance and insurance and utilities sectors exhibited higher adoption compared to organisations from other sectors, and technical practices are adopted at a higher level compared to management and support practices. The survey also found that small firms were interested in software process improvement.

The process improvement assessments of 22 firms confirmed that the capability of technical processes is higher than that of management processes; and suggested that higher capability is associated with the proportion of experienced staff, the proportion of qualified staff, firms undertaking projects with lengthy durations, and certification to ISO 9001.

The evaluation of the PIP was conducted by analysing the 22 organisation context forms and assessment reports, and the 20 reports from the follow-up meetings. The evaluation revealed that assessment-based SPI programs are effective for small firms, regardless of the maturity of the processes at the time of the assessment. The extent of adoption is associated with the proportion of formally qualified staff, the proportion of technical staff, and with shorter time periods from assessment to follow-up. Consideration of the impact of the critical success factors from the literature with the analysis of the PIP reports suggests that factors that impact large firms are not the same as those that impact small firms.

Finally, the comparison of the results and findings of the best practice survey and the process improvement reports highlighted inconsistencies between survey and PIP results, and confirmed poor adoption of practices by small firms, in particular non-technical processes.

The next chapter concludes the thesis. It provides implications for theory and practice, discusses the limitations of the study, and suggests areas requiring further research.

9 CHAPTER NINE—CONCLUSIONS AND IMPLICATIONS

9.1 Introduction

This is the final chapter, and after summarising the preceding work, the conclusion about the research problem and the contribution of the research to the body of knowledge is presented. Then implications of the research for theory, practice and policy are explored. The limitations of the research method and analysis are discussed in §9.5, followed by areas for future research suggested in §9.6.

9.2 Summary and conclusion

In this section, a summary of each of the preceding eight chapters is provided, the findings are stated and the research questions answered. The conclusion to the research problem is stated, and the contributions of this research are discussed.

9.2.1 Summary of the research

The first chapter provided the foundation for the study. Firstly, background information describing the Australian software development industry was provided and the significance of the research was established. This study was motivated by the call for empirical research on software process innovation (Fenton 1993a; Fenton, Pfleeger & Glass 1994; Jung et al. 2001; Mustonen-Ollila & Lyytinen 2003), and in particular, the need for evaluation of SPI programs (Goldenson et al. 1997; Hersh 1993). In addition, the study was a response to the requirement to collect and disseminate improved industry statistics and undertake benchmarking of the local software industry (Goldsworthy 1997). The research problem was stated in Chapter 1:

> *To what extent are best practice techniques used by Queensland software development organisations, and how effective are assessment-based SPI programs for small Australian software development firms?*

Three research questions were defined to explore the research problem, the research was justified on theoretical and practical grounds, the scope of the research was defined and the key assumptions stated.

Chapter 2 reviewed literature related to TQM, diffusion of process innovation and organisational maturity before summarising current research about software process improvement. From this literature review, it is apparent that there is little research relating to SPI in small organisations. Based on the literature, hypotheses were formulated to provide answers to the research questions posed in Chapter 1.

As described and justified in Chapter 3, this study used a positivist multi-method research approach. Survey and field experiments were selected as the best approach to explore the research problem and to answer the research questions developed to guide the project.

In Chapter 4, the survey instrument and procedures were described, and the justification for the field experiments provided. Limitations of the survey method and ethical issues related to the survey were discussed.

After presenting a summary profile of the 203 survey respondents, Chapter 5 explored the distribution of the variables representing the extent of adoption of best practice, and their associations with organisational characteristics. A summary of the results of the hypothesis tests relating to the survey was provided.

The findings from the best practice survey can be summarised as follows:

- Queensland organisations involved in software development have adopted, on average, almost half of the best practices.
- COTS developers show significantly higher levels of adoption compared to non-COTS developers for 23 percent of the practices (10 of 44 practices).
- Practices related to the primary systems development lifecycle tend to be adopted at a higher rate compared to the management and support practices.
- Generally, higher levels of adoption were found for practices relating to the involvement of stakeholders such as senior management and customers, and project estimation. Poor adoption was reported for practices relating to project monitoring and control, and metrics.
- The gap between the leaders in adoption of best practice and the laggards is extreme. The leaders have significantly higher levels of adoption compared to the laggards in 90 percent of the practices (40 of the 44 practices).
- Larger software development groups show higher levels of adoption compared to smaller software development groups.

- The overall adoption of development organisations from Queensland compares favourably with developers from European countries.
- Higher adoption of best practice was found in the industry sectors of finance and insurance, utilities, government administration and defence, software development and IT. Lower adoption was evidenced in the mining and education sectors.
- Aspects of software development with the most scope for improvement include technical issues, specifically testing and the need for automated support, especially for configuration management; project management; and quality assurance.

In Chapter 6, the field experiments, comprising 22 process capability assessments, were detailed in terms of the selection of subjects, the design, and protocol of the field experiments, and the compilation of results. The variables were defined and data analysis method described. Limitations to the field experiment methodology and ethical considerations such as confidentiality were discussed.

Chapter 7 focused on the analysis of the 22 assessment reports and 20 final reports, and supporting documents such as the organisation context forms. After summarising the organisation demographics of the 22 firms, the relationships between the capability of the processes and demographics were evaluated. The extent of process improvement from the time of the initial assessment to the follow-up meeting was analysed and compared with selected organisation characteristics. The data from the survey was then compared with that gathered through the field experiments. Finally, a summary of the results of the related hypothesis tests was provided.

The findings from the analysis of the PIP assessments are summarised as follows:

- One third of all the processes were rated at level 0 and almost half were rated at level 1.
- The capability of primary lifecycle processes was rated higher than that of support processes. Requirements elicitation and software development processes were rated with highest capability levels, and process establishment recorded the lowest capability level.

- Higher process capability is associated with effective staffing, specifically with the proportion of experienced staff, the proportion of staff with formal education, and the proportion of technical staff.
- Higher process capability may be associated with certification to ISO 9001, but as there were only two certified firms, this conclusion is treated with caution.
- Higher process capability is associated with longer project durations.

Chapter 8 interpreted the findings from Chapters 5 and 7 in terms of the three research questions. Qualitative analysis of the assessment and final reports was used to explore issues related specifically to the assessed firms, and the PIP method. Critical success factors identified in previous SPI studies were then discussed in the context of this study to highlight factors of particular relevance to small development firms.

The findings from the analysis of the PIP follow-up meetings are summarised as follows:

- 15 firms reported improvement; six of these firms achieved higher process capability levels.
- Across the program, all processes showed capability level improvement, but the process showing the greatest improvement was process establishment, and the least improvement was seen in requirements elicitation.
- The extent of improvement is associated with the proportion of technical staff and the proportion of formally qualified staff.
- Greater improvement in process capability is associated with a shorter period of elapsed time from the initial assessment to the follow-up meeting.

The comparison of the findings from the best practice survey with the process improvement program revealed that best practice survey respondents over-estimated their use of best practice, and that the higher adoption of technical practices compared to supporting practices is consistent with the finding of higher capability of technical processes compared to support processes.

Research question 1. Is there wide variation in the extent of adoption of software development best practice techniques by software developers in Queensland, and is the extent of adoption related to particular organisational characteristics?

The study confirmed that there is wide variation in terms of the adoption of individual practices, as well as the extent of adoption of best practice by organisations. While project management planning and customer involvement practices are widely adopted, the use of metrics for estimating and testing are barely used by the organisations which responded to the survey. Overall, practices of a technical nature are more widely adopted compared to techniques related to support and management. COTS developers have higher adoption than firms which do not develop COTS systems and adoption is associated with the size of the development group. The leaders in adoption have significantly better practices when compared to the laggards for 40 of the 44 practices included in the survey. Furthermore, organisations from the finance, insurance and utilities sectors exhibited higher adoption compared to organisations from other sectors.

Research question 2. Is the capability of technical processes higher than that of management processes, and is process capability associated with particular organisational characteristics?

The process improvement assessments of 22 firms confirmed that the capability of technical processes is higher than that of management processes; and suggested that higher capability is associated with the proportion of experienced staff, the proportion of post graduate qualified staff, firms undertaking projects with lengthy durations, and certification to ISO 9001.

Research question 3. Are assessment-based SPI programs an effective means to improve process capability for small software development firms, and is the extent of improvement associated with particular organisational characteristics?

The evaluation of the PIP was conducted by analysing the 22 assessment reports, and the 20 final reports from the follow-up meetings. The evaluation revealed that assessment-based SPI programs are effective for small firms, regardless of the maturity of the processes at the time of the assessment. The extent of improvement is associated with the proportion of technical staff and formally qualified staff.

9.2.2 Conclusion and contributions

This research has determined the extent of use of best practice by software developers in Queensland, and then empirically assessed the effectiveness of a rapid process improvement intervention in small software development firms. The objective was to understand software development processes being used, and to evaluate an assessment-based software process improvement initiative undertaken by 22 small firms. In investigating the research problem, this study found that Queensland software development organisations employ limited use of best practice techniques, and that assessment-based SPI programs are effective for small software development firms. This study has made numerous and varied contributions which are now listed in decreasing order of importance.

Contribution #1 This study has detailed the experience and outcomes of a process improvement program conducted with 22 small software development firms. The evaluation established that as well as providing a powerful way to diffuse awareness of software best practices, the program constituted an effective improvement path for small firms.

The in-depth evaluation revealed that although only six of the 22 firms increased in terms of capability levels, a further nine firms reported improvements to their development processes. The analysis of the PIP established that the program provided a practical introduction to formal SPI and provided a solid foundation for further SPI initiatives. This conclusion challenges the view that SPI is not feasible unless the firm has visible, defined processes. As detailed in §8.5.1, 11 of the 17 firms with low initial capability benefited from the RAPID assessment and exhibited improvements. The success of this program indicates that SPICE-based RAPID assessments are an appropriate way for small organisations to get started on SPI.

The most valuable contribution of this research is the evaluation of the RAPID method which provides a realistic option for small development organisations which traditionally lack the resources to undertake full-blown software process assessments.

With the Australian software community dominated by very small organisations (88 percent have less than five staff), this method may provide valuable opportunities for such developers to evaluate and improve their processes, thereby achieving success in domestic and global markets.

Contribution #2 As well as detailing the capability of 22 small software firms, this study provides an interesting insight into the actions, reasons for inaction, and reactions of the firms as far as implementing the recommendations from the assessments. Analysis of the reactions of the participants of this program confirms the importance of mentoring, training and organisation stability, but challenges the relevance of senior management support as a critical success factor for small firms.

This research answers the call to reduce the scepticism and uncertainty which exists in relation to the accuracy and usefulness of software process assessments and improvements based on them (Goldenson et al. 1997). Furthermore, although there are many published accounts of assessments, there is little reported about reappraisals or follow-ups except for those involving large high maturity organisations, and even less has been reported about the SPI experience of Australian firms. Moreover, valuable experience has been obtained and documented to further disseminate SPI within the software industry.

Contribution #3 As reported in Chapters 4 and 5, an extensive survey of the Queensland software industry was conducted to identify and compare *best practice* in software development with *current practice*. Despite some reservations stated above (§8.2.8) about the ESI questionnaire, the survey has achieved its goal of providing benchmark information about adoption of best practice by software developers in Queensland and provided a much better understanding of the practices used by small software development firms.

Contribution #4 As far as is known, this is the only study that has investigated and established that firms developing commercial off-the-shelf packages have higher rates of adoption of best practice compared to firms that do not develop COTS systems. As the PIP assessment reports did not clearly indicate which of the firms were developing

COTS systems, it was not possible to compare the process capability of COTS and non-COTS firms.

Contribution #5 The overall adoption of 48 percent implies that the organisations which responded have adopted, on average, almost half of the best practices in the SBPQ, indicating scope for improvement in adoption. While this overall adoption rate places the Queensland software industry in a competitive position compared to adoption of firms in European countries, there is scope for improvement.

Contribution #6 A valuable outcome of this study is the compilation of the list of critical success factors for small firms, suggesting that the factors that inhibit and enable SPI for large organisations are not the same as those that affect SPI for small organisations. For small firms, senior management takes an active role, therefore commitment is guaranteed. Organisation politics and communication issues are typical in large firms, but not critical in small organisations. Small firms need a low cost SPI program with a restricted scope, a short time frame to evaluation, and mentoring from external assessors/consultants. It is also crucial that the firm is not disrupted by internal or external events during the course of the SPI program. Furthermore, this study provides a contribution to assessment methods by providing recommendations to improve the RAPID method.

Contribution #7 As well as providing a profile of the software industry in Queensland, and the difference in adoption of various practices by COTS and non-COTS developers, the survey provided SEA (Qld) and SQI with a large target list of up-to-date contacts. This contact list is a valuable resource for planning and organising assessments, training courses and other elements of the National Industry Improvement Program. The survey also revealed areas the software developers wished to improve and also suggested how SEA could help provide the firms' needs. Every survey respondent was provided with the URL and password to access a web-based summary of the survey results that highlighted the organisation's response to each question. This enabled respondents to benchmark their own development practices against the aggregated responses, thereby highlighting their strengths and weaknesses.

Contribution #8 The PIP part of the study added to the methodology of evaluating the outcomes of assessments in two separate ways. Firstly by using two measures: capability level and attribute achievement. The use of the second measure of capability revealed that although only six firms improved capability levels, one other firm improved in capability by raising its attribute achievement rating. In general, the attributes corresponding to the higher capability levels less often receive higher ratings compared to the attributes corresponding to the lower capability levels. Less than ten percent of process ratings failed to achieve a particular capability level because of inadequacies in the previous level, rather than the level in question (§7.7.1). This result supports the level hierarchy of the SPICE model. Thirdly, some of the hypothesis tests provided different results for the two measures. For some processes, where the association of the capability level with an organisation factor was not significant, it was significant for the attribute achievement, thereby signally further research is required. This inconsistency would not have been apparent if only one measure of process capability had been used.

The other way in which this study added to the methodology of evaluating assessments was by performing qualitative analysis of the assessment and final reports, in addition to the quantitative analysis of the process profile data. This qualitative analysis summarised advice provided by assessors, and extracted common themes across the reports from the 22 firms, thereby providing rich insights into the improvement process.

Related publications from the study

The results and experiences of this study have been presented to the software industry and the software engineering research community in Australia and internationally. These papers were peer reviewed and the reviewers' comments prompted the researcher to undertake further analysis of various aspects.

Early results of the survey were presented to the Software Engineering Australia conference (Cater-Steel 1999). After further analysis, the survey results were presented to the Australian Software Engineering Conference (Cater-Steel 2000b). The survey findings were compared with a subset of the assessment results at the European Conference on IT Evaluation in Dublin (Cater-Steel 2000a), and that paper was revised and subsequently published in the *Electronic Journal of Information Systems Evaluation* (Cater-Steel 2002b). A paper detailing the process assessments of four firms was presented to the Australian Software Engineering Conference (Cater-Steel 2001), and the discussion was extended and presented to the International Conference on Software Engineering and Applications in Cambridge, Massachusetts (Cater-Steel 2002a). A detailed description of the outcomes of the process improvement program was presented to the Australian Software and Systems Engineering Process Group Conference in Surfers Paradise (Cater-Steel 2003). The response by low capability firms was the focus of a paper presented to the Australian Software Engineering Conference (Cater-Steel 2004). Finally, the qualitative analysis of the outcomes of the assessments was presented to the International SPICE Conference on Process Assessment and Improvement at Lisbon (Cater-Steel, Toleman & Rout 2004).

9.3 Implications for theory

This study has drawn on reference disciplines from outside the software engineering field for theory and concepts, thereby challenging Jarvinen's (2000) assertion that 'there is no sound basis for the concepts of software process maturity and capability, except for some loose analogy to SPC [statistical process control]' (p. 76). This study has integrated the concepts of TQM, in particular process improvement, with research in the field of software process improvement. Broad theories of management, and specifically theories relating to small firms, have been applied in this research to further understand the adoption of best practice and the outcomes of process improvement by software development organisations.

As explained by El Emam and Birk (2000b), the benchmarking paradigm is commonly used for improving software engineering practices, and best practices become codified into an assessment model such as ISO 15504. This view of practice leading to theory is shared by Glass (1996), who gives examples from aerodynamics and thermodynamics. This research has contributed to the theory of SPI by providing validation for the RAPID model. As the RAPID model is based on the SPICE standard, this research has also provided validation of the emerging international standard. It is essential that the theory underlying such models is confirmed by empirical research (Fenton, Pfleeger & Glass 1994; McBride 2004). A further implication for theory concerns the utility of the RAPID model. This research has established that the RAPID method is an effective means for small low capability firms to undertake software process improvement.

There have also been concerns about the lack of theoretical support for staged models such as the SE-CMM (Bollinger & McGowan 1991; Drehmer & Dekleva 1993). This research provides empirical support that firms do in fact follow an evolutionary progression of process improvement, consistent with the sequence of the SPICE hierarchy of capability levels.

9.4 Implications for practice and policy

This study has implications for all stakeholders in the software industry: developers, managers, researchers, consultants and clients. The first part discusses the effect of the study findings on software developers, firms and consultants (§9.4.1). Public policy is also considered: the government sector is the largest software purchaser, and government policy impacts on the software industry (§9.4.2). The third section considers implications generally in relation to purchasers of software (§9.4.3).

9.4.1 Software developers, firms and consultants

Prior to providing recommendations, aspects of the professionalism of software developers are discussed, followed by an important issue for firms: competitive advantage.

Software professionals

From the survey, the overall adoption of 48 percent implies that the organisations have adopted, on average, almost half of the best practices in the SBPQ, indicating scope for improvement in adoption. The variation in the adoption of practices indicates that some developers or their managers believe that most of the practices included in the questionnaire are essential, whereas others are either unaware of such practices, or consider such practices to be of little value. This inconsistency in practice has contributed to inconsistent project outcomes and has given software development a bad reputation (Birmingham 1996; Sweeney Research 2003b). In analysing the process capability of the PIP firms, this study considered the proportion of technical staff, the proportion of experienced staff, and the proportion of formally educated staff in each firm. Some aspects of staff professionalism were found to be associated with process capability and also the extent of process improvement.

Professional licensing and certification has ensured professions such as accountancy, medicine and law enforce standards of best practice with recognised qualifications. Unlike other industries with uniform best practice, the software industry does not have universally accepted practices. This study has highlighted the vast gap in adoption of best practice between the leaders and laggards. A number of Australian studies have stressed the need for software developers to be accredited (Goldsworthy 1997; McKerlie Consulting 1996; Sweeney Research 2003b). Could this variation be overcome by professional certification or licensing of software developers?

Certification is adopted by some professions, for example in the accounting field in Australia, many accountants are Certified Practicing Accountants (CPA). Certification is a voluntary process administered by a professional society, and usually extends beyond a limited geographic area to national or sometimes even international regions (McConnell 2003). Licensing, on the other hand, is typically administered by jurisdictions and involves a mandatory legal process that is intended to protect the public (McConnell 2003).

The Australian Computer Society (ACS) promotes a professional certification scheme named Practising Computer Professional, but it is not widely adopted or legally enforced in the industry. Since 1992, the IEEE Computer Society has promoted the software engineering fundamentals-based certification: Certified Software Development Professional (CSDP) (Seidman et al. 2003) which is largely aligned with the scope of the SWEBOK guide (Abran et al. 2001). Recently, there has been an increase in the awareness and popularity of exam-based software industry certification such as Microsoft certified professional, Novell certified network engineer, and Oracle certified professional (Framework for the Future Mapping Working Group 2002). Perhaps if SEA and ACS promoted awareness of the CSDP as an international certification, then software developers would apply best practice and improve the quality of project outcomes.

Competitive advantage

It has been reported (Loane 2003) that only 25 per cent of the A$4 billion software sold in Australia each year is developed by local firms. Furthermore, it has been estimated that small-medium enterprises would win a minimal proportion, between five and ten percent, of the A$3.5 billion spent by the Federal Government on ICT in 2003 (IDG Staff 2003). Large Australian software purchasers, such as the Defence Materiel Organisation (DMO), Telstra and ANZ Bank, are moving towards international standards such as CMMI and SPICE (Howarth 2004b, 2004a; Marshall & Hofmann 2001). In their consideration of diffusion theory, Bayer and Melone (1989) argue that mandated software engineering innovations first introduced to a government contractor population will later transfer to the commercial sector because members of one population interact with, and in fact may jointly belong to, the other population. CMMI is also gaining international acceptance in the software engineering community: of the 87 CMMI appraisals performed up to mid 2003, only 39 were carried out in the USA (Phillips 2003).

CMMI was developed to be consistent and compatible with ISO 15504 (Chrissis, Konrad & Shrum 2003). Mapping between CMMI and SPICE is available (Rout, Tuffley & Cahill 2001) and compliance with one standard affords compliance to the other. Therefore, small firms starting with a SPICE-based RAPID assessment can proceed towards CMMI recognition.

So in order to gain a greater share of the domestic and international market, small software firms need to implement recognised SPI programs such as the one discussed in this study. Software process assessment proves to investors and customers that the firm is committed to software quality (Saran 2001). Increasingly, large companies, such as Telstra, are recognising CMMI benchmark results when selecting their suppliers (Howarth 2004a). Furthermore, with the increasing trend of large Australian organisations outsourcing their software development to firms in countries such as India, it is critical for the local software industry to adopt SPI to improve quality and productivity. For small software firms, the RAPID assessment method provides a gentle introduction to the experience of adopting international standards whilst providing a practical and effective method to assess and improve current processes.

The low adoption of best practices, as indicated from the survey, suggests that process improvement should be a high priority for many development firms. The survey also established that many firms are interested in improving their processes. This study has demonstrated that the SPICE-based low-rigour RAPID assessment is an effective method to achieve improvement. The implication of this study for small firms is that if they can undertake a RAPID assessment, it may help them gain a bigger share of government spending and also provide a competitive advantage in domestic and international markets.

Recommendations to small development firms

From the analysis of the current literature and also the assessment and final reports, the following recommendations are made to assist small firms undertaking SPI, and also assessors involved in such projects:

- Before commencing SPI, ensure the organisation is stable and not undergoing major disruptions from internal or external events.
- Firms should draw on expertise of external assessors/consultants as mentors.
- The SPI action plan, derived from the assessment recommendations, should be realistically achievable within the evaluation time-scale.
- Plan the evaluation from the start of the SPI program, this will be a source of motivation.
- Ensure that managers and development staff receive adequate training specific to the SPI model and areas of improvement.

Recommendations to SEA and consultants

In compiling the census list of all companies involved in software development and recording survey responses, the survey provided SEA (Qld) and SQI with a large target list of up-to-date contacts. This list is a valuable resource for planning assessments, training courses and other elements of the National Industry Improvement Program. As small firms generally do not have internal SPI expertise, and find consultants' costs prohibitive, improvement programs need to be subsidised by industry bodies (such as SEA) or university research centres (such as SQI). In order to obtain sufficient quantity of material for research, it is necessary to have assistance from consultants (Voss et al. 1998). As small firms find SPI costly, such a program needs to be substantially subsidised.

As discussed in §8.5.2, the evaluation of the PIP has highlighted areas of improvement and the following recommendations (to assessors/consultants) are made to improve RAPID assessments:

- Provide detailed information to the sponsor about the method and model prior to the assessment.
- Assessors should meet the sponsor prior to the assessment, not just plan by phone/email. Need to nurture a relationship of confidence and trust.
- Ensure that the follow-up assessor is one of the initial assessors.
- Redesign the organisational context questionnaire to provide clearer instructions. This will ensure more accurate organisation and project data can be collected.
- Include a template for sponsors to record all costs and benefits from the time of the initial assessment to the follow-up assessment.
- Provide documented guidance to the follow-up assessors for the procedure for the follow-up meetings.
- Devise a feedback form for the sponsor to complete at the time of the follow-up meeting.
- During the time period from the initial assessment to the follow-up assessment, encourage the assessor to contact the sponsor at least on a monthly basis to providing ongoing support and develop trust.

9.4.2 Public sector policy analysts

There are two main issues related to public sector policy. The first issue concerns mandates in the acquisition policies of state and federal governments; the second issue relates to government support for the local industry in terms of funding to industry bodies (for example, SEA; ACS) and research centres such as universities. Although the most successful process for the propagation of best practice is through the supply chain, it is better if industry groups consciously plan site-to-site mentoring (Voss et al. 1998, p. 16).

In a recent report on the Australian software industry, more than half the respondents expressed concern about being excluded from government tenders due to requirements for compliance to standards such as CMMI (Sweeney Research 2003b, p. 17). For example, the Defence Materiel Organisation (DMO) will use CMMI appraisals during offer definition and/or contract surveillance on all high value, high risk acquisitions (Department of Defence 2004, p. 77).

The Sweeney research also revealed that three quarters of firms believe that government should play a greater role in supporting the Australian software industry (2003b, p. 3). It has been suggested that the most obvious way for the government to assist the software development industry would be to 'buy Australian' (McKerlie Consulting 1996, p. 42; Woodhead 2003). Currently, it is very difficult for small firms to get a share of government ICT spending (Davidson 2004). Nathan Brumby, CEO of SEA (Australia) has called on the Federal Government to follow the lead of other countries such as Israel, the United States and Ireland with tax strategies to favour the local software industry (Woodhead 2003).

9.4.3 Software purchasers

Some large non-government organisations apply SPICE or CMMI for supplier selection, for example, Telstra and the ANZ Bank. Also, Queensland's largest private corporation, Suncorp, relies on SPICE assessments to improve its in-house software development processes (Bullen 2003). It is expected that other purchasers will follow this trend and require process capability appraisals, especially in relation to systems considered high risk or high value.

9.5 Limitations of the research

As with any study, there are limitations to this research. While some of the limitations must be accepted, as little can be done to overcome them, others may be addressed in future research (in §9.6). Earlier in §1.8, limitations of the research were outlined. In the interests of practicality, the scope of the research has been limited to small-medium Queensland software development firms. It is recognised that such firms may have cultural and locational-driven characteristics which distinguish them from counterparts in other states or countries.

Survey limitations

In addition to limitations relating to the survey data collection methods and data analysis procedures as discussed in §4.5, several other limitations became apparent during the survey stage of the research. The poor response rate may have been improved by offering an incentive such as a voucher for a training session at SEA. Rather than attempt a census of the entire industry, the use of a sampling frame combined with more persistent follow-up of non-respondents may also have been more effective. In §8.6.1, concerns are raised relating to the accuracy of the survey responses. Perhaps, rather than *yes/no* options for the adoption questions, the provision of a scale, for example *all projects; for some projects; no/not applicable; don't know*, may have improved the precision.

PIP limitations

Known limitations of the PIP field experiments are discussed in §6.12. As well, the outcomes of the PIP for some of the firms were affected by internal and external events beyond the control of the researcher, such as Y2K, GST, management restructuring, firm relocation. As discussed in §3.4.2, it is neither realistic nor practical to set up a control group in an industry experiment such as the PIP. In the absence of a control group, the research design uses the initial assessment as the control, and assumes that the capability levels would have remained unchanged if the program had not been conducted. It is recognised that some firms may have improved their processes regardless of their participation in a formal SPI program.

Although the researcher was involved in three of the assessments, the main effort of the evaluation of the PIP involved analysing work performed by other people. The limitations associated with this approach included the unavailability of some of the feedback forms (discussed in §6.12.1), and the poor measurement accuracy of the organisation context questionnaire.

Limitation in statistical analysis of 22 firms

The major data collection effort for the field experiments involved only 22 firms, and of these, only 9 had formal reassessments. As a result, simple correlation and comparison techniques were used to determine relationships between capability and other factors. Thus further research with larger samples is needed to confirm the study's findings.

Generalisation from 22 firms

Another limitation has to do with the extent to which the findings can be generalised beyond the 22 firms studied. The number of cases is too limited for broad generalisation, but the field experiments represent a variety of outcomes of SPI. Therefore, software development firms undertaking SPI activities can benefit from the findings. Further empirical evaluations are needed to replicate the findings in different contexts and surroundings.

9.6 Future directions in research

As mentioned in previous chapters, this work has opened many possibilities for further research. The limited application of Hofstede's national cultural scores in this research could be extended to investigate the impact of national culture on adoption of best practice and software process improvement. With the increasing use of packaged software and components, the finding that COTS-developers have adopted higher levels of best practice compared to non-COTS developers needs to be explored in other Australian states and also across the international software development community. Also, it has been suggested that in-house development groups are lagging in their adoption of best practice compared to commercial development firms, although little research to date has addressed this issue.

Future research is needed to explore the role of mentors. In the *Showcase* program, not reported in this study, funding was provided to SQI to conduct full SPICE assessments and mentoring for two firms. A detailed comparison of the outcomes of the *Showcase* program with the 20 reported in this study may clarify the importance of mentoring for small firms undertaking SPI. Furthermore, recognising the prohibitive cost of SPI assessments for small firms, future research could examine the feasibility of a self-assessment instrument.

Further empirical evaluations are needed to replicate the PIP field experiments in different contexts and surroundings, such as other Australian states, or using other rapid assessment methods. A longitudinal study would help determine the optimal time period from assessment to reappraisal. Extended longitudinal data would confirm the effectiveness of the program in the long term. For example, did the firms continue to improve their processes? Did the SPI program result in benefits in terms of improved quality, productivity and reduced defects? How does SPI impact on business success or failure?

The study reported in this thesis was based on ISO/IEC TR 15504. A major issue for future research involves the evolution of the ISO 15504 standard. In response to feedback from the SPICE trials and ballots, the structure of the 15504 standard has undergone a dramatic change from nine parts in the Technical Report version, to five parts in the Final Draft International Standard (FDIS) version, with a notable change being the removal of the process dimension (Rout 2003). To compensate for this, ISO/IEC 12207 (ISO/IEC 2002) has been amended to include more detail about process objectives and outcomes. When the FDIS version is released, predicted to be in the mid-part of 2005, it is expected that interest in the ISO 15504 standard will grow, from both software development firms, and from organisations acquiring software. Some firms currently using ISO/IEC TR 15504 are already planning how they will manage the transition to the FDIS version. The widespread adoption of the international standard will see increased interest in the value of training courses and mentoring, issues barely addressed in this study.

Another area of future research concerns how the growing popularity of the CMMI will impact on the future adoption of ISO 15504. Although the assessment profiles can be mapped from one model to the other, the preference by some industries or governments will impact on the future of both process capability models.

9.7 Concluding remarks

In summary, this research established the extent to which software developers in Queensland are using best practice, and has empirically assessed the effectiveness of a SPICE-based software process improvement program in small software development firms. To achieve this objective, an extensive survey of the software industry compared *best practice* in software development with *current practice* in Queensland organisations. This research has determined that there is wide variation in the adoption of software development practices, and that low-cost SPICE-based assessments are effective for software process improvement for small development firms, including those firms with low initial capability. Following on from this, a detailed evaluation of a software process improvement initiative in 22 small firms was carried out, providing insights into the adoption of best practice by Queensland software development organisations, and the association of organisation and project characteristics with process capability and the extent of improvement.

Through applying management and software engineering theories, this research has advanced the position of previous research resulting in a clearer understanding of the critical success factors of SPI for small firms. Furthermore, it has provided validation for the RAPID model and method. As a result, this research has made several contributions to theory and practice. In addition, this research presents opportunities for future research about national culture issues, the role of mentors, and the evolution of ISO 15504.

The objectives of this research have been achieved, and insights have been provided into current practice and process improvement in small firms. The dissemination of this research will better equip practitioners and consultants to undertake software process improvement, hence increasing the software development firm's potential for success.

Bibliography

Abrahamsson, P 2001, 'Rethinking the concept of commitment in software process improvement', *Scandinavian Journal of Information Systems*, vol. 13, pp. 69-98.

Abran, A, Moore, JW, Bourque, P, Dupuis, R & Tripp, L (eds) 2001, *SWEBOK: Guide to the Software Engineering Body of Knowledge: Trial Version*, IEEE Computer Society, Los Alamos.

ABS 1993, *Australian and New Zealand Standard Industrial Classification (ANZSIC).Classification titles and codes*, viewed 2003 6 July, <http://www.abs.gov.au/ausstats/ABS@.nsf/0/6741fc1736a81aefca25697e0018fb59?OpenDocument>.

---- 1995, *Small Business in Australia*, 1321.0, Canberra.

---- 2002, *AusStats: 8126.0 Information Technology, Australia*.

---- 2004a, *8127.0 Characteristics of Small Business*, Australian Government, viewed 8 Sept 2004, <http://www.abs.gov.au/ausstats/abs@.nsf/0/E49E3B4DC3595C92CA2568A900139377?Open>.

---- 2004b, *8126.0 Information and Communication Technology, Australia 2002-03*, viewed 15 Oct 2004, <http://www.abs.gov.au/Ausstats/abs@.nsf/1020492cfcd63696ca2568a1002477b5/60c99311996ef774ca256f070079c995!OpenDocument>.

Adam, EEJ 1994, 'Alternative quality improvement practices and organisation performance', *Journal of Operations Management*, vol. 12, pp. 27-44.

Agarwal, R & Prasad, J 2000, 'A field study of the adoption of software process innovations by information systems professionals', *IEEE Transactions on Engineering Management*, vol. 47, no. 3, pp. 295-308.

Agresti, WW 1993a, ‘Session 1 summary: the experimental paradigm in software engineering’, in HD Rombach, V Basili & RW Selby (eds), *Experimental Software Engineering Issues: critical assessment and future directions: Proceedings International Workshop 1992*, Springer-Verlag, Dagstuhl Castle, Germany, pp. 33-7.

---- 1993b, ‘Profile of an artefact assessment capability’, in HD Rombach, V Basili & RW Selby (eds), *Experimental Software Engineering Issues: Critical Assessment and Future Directions*, Proceedings International Workshop 1992, Dagstuhl Castle, Germany edn, Springer-Verlag, pp. 13-6.

Ahern, DM, Clouse, A & Turner, R 2001, *CMMI® Distilled: A Practical Introduction to Integrated Process Improvement*, 2 edn, SEI Series in Software Engineering, Addison Wesley Professional.

Alavi, M & Carlson, P 1992, ‘A review of MIS research and disciplinary development’, *Journal of Management Information Systems*, vol. 8, no. 4, pp. 45-62.

Anacleto, A, von Wangenheim, CG, Salviano, CF & Savi, R 2004a, ‘Experiences gained from applying 15504 to small software companies in Brazil’, in *Proceedings of the International SPICE Conference on Process Assessment and Improvement*, Lisbon, Portugal, pp. 33-7.

---- 2004b, ‘A method for process assessment in small software companies’, in *Proceedings of the 4th International SPICE Conference on Process Assessment and Improvement*, Lisbon, Portugal, pp. 69-76.

Anderson, M & Sohal, AS 1998, *A Study of the relationship between quality management practices and performance in small business*, Dept of Management Working Paper Series 77/98, Faculty of Business and Economics, Monash University.

Applegate, LM, Austin, RD & McFarlan, FW 2003, *Corporate Information Strategy and Management: Text and Cases*, 6th edn, McGraw-Hill, New York.

Ares, J, Garcia, R, Juristo, N, Lopez, M & Moreno, AM 2000, 'A more rigorous and comprehensive approach to software process Assessment', *Software Process: Improvement and Practice*, vol. 5, pp. 3-30.

ASEI 1996, *Software Development in South Australia*, Australian Software Engineering Institute.

ASQ 2003, *Walter A. Shewhart: Father of statistical quality control*, viewed 3 Mar 2003, <http://www.asq.org/join/about/history/shewhart.html>.

Attewell, P & Rule, JB 1991, 'Survey and other methodologies applied to IT impact research: experiences from a comparative study of business computing', in K Kraemer (ed.), *The Information Systems Research Challenge: Survey Research Methods*, vol. 3, pp. 299-315.

Australia Post 1999, *PrintPost Sort Plan*, viewed 2 Feb 1999, <http://www1.auspost.com.au/download/PrintPost_sort_plan.pdf>.

Azuma, M 1995, 'Towards the 21st Century's software, state of the art and international standards in JTC1/SC7', in M Lee, B Barta & P Juliff (eds), *Software Quality and Productivity: Theory, Practice, Education and Training*, Chapman & Hall, pp. 3-9.

Azuma, M & Mole, D 1994, 'Software management practice and metrics in the European Community and Japan: some results from a survey', *Journal of Systems Software*, vol. 26, pp. 5-18.

Bach, J 1995, 'The challenge of "Good Enough" software', *American Programmer*, vol. 8, no. 10, pp. 2-11.

Baddoo, N & Hall, T 2002a, 'Practitioner roles in software process improvement: an analysis using grid technique', *Software Process Improvement and Practice*, vol. 7, no. 1, pp. 17-31.

---- 2002b, 'Motivators of software process improvement: an analysis of practitioners' views', *The Journal of Systems and Software*, vol. 62, pp. 85-96.

---- 2003, 'De-motivators for software process improvement: an analysis of practitioners' views', *The Journal of Systems and Software*, vol. 66, pp. 23-33.

Bamford, RC & Deibler, WJ 1998, 'Hybrid multi-model assessment: when the CMM meets ISO9001', *Crosstalk: The Journal of Defense Software Engineering*, Sept.

Bandinelli, S, Fuggetta, A, L., L, Loi, M & Picco, GP 1995, 'Modeling and improving an industrial software process', *IEEE Transactions on Software Engineering*, vol. 21, no. 5, pp. 440-54.

Barron, JM & Paulson Gjerde, K 1996, 'Who adopts total quality management (TQM): theory and an empirical test', *Journal of Economics & Management Strategy*, vol. 5, no. 1, pp. 69-106.

Basili, VR 1993, 'The experimental paradigm in software engineering', in HD Rombach, V Basili & RW Selby (eds), *Experimental Software Engineering Issues: critical assessment and future directions, proceedings International Workshop 1992, Dagstuhl Castle, Germany, LNCS 706*, Springer-Verlag, pp. 3-12.

---- 1996, 'The role of experimentation in software engineering: past, current, and future', in *Proceedings of the 18th International Conference on Software Engineering*, IEEE Computer Society Press, pp. 442-9.

Basili, VR & Green, S 1994, 'Software process evolution at the SEL', *IEEE Software*, vol. 11, no. 4, pp. 58-66.

Baskerville, R & Wood-Harper, AT 1998, 'Diversity in information systems action research methods', *European Journal of Information Systems*, vol. 7, pp. 90-107.

Batista, J & Dias de Figueiredo, A 2000, 'SPI in a very small team: a case with CMM', *Software Process: Improvement and Practice*, vol. 5, pp. 243-50.

Bawden, V 1994, 'Software Quality Certification: A Rational Client's Perspective.' MIT thesis, UQ.

Bayer, J & Melone, N 1989, 'A critique of diffusion theory as a managerial framework for understanding adoption of software engineering innovations', *Journal of Systems & Software*, vol. 9, no. 2, pp. 161-6.

Beecham, S, Hall, T & Rainer, A 2001, *Software process improvement in twelve software companies: an empirical analysis*, SCB/Problem Paper, Department of Computer Science, University of Hertfordshire.

Bengtsson, L, Elg, U & Lind, J-I 1997, 'Bridging the transatlantic publishing gap: how North American reviewers evaluate European ideographic research', *Scandinavian Journal of Management*, vol. 13, no. 4, pp. 473-92.

Benson, T, Bull, A & Standen, D 1999, *The IT Engine Room: SMEs in Australia's IT&T industry*, Department of Communications, Information Technology and the Arts, Australian Information Industries Association, Commonwealth of Australia.

Bickerdyke, I, Lattimore, R & Madge, A 2000, *Business Failure and Change: An Australian Perspective*, Ausinfo, Canberra.

Billings, C, Clifton, J, Kolkhorst, B, Lee, E & Wingert, WB 1994, 'Journey to a mature software process', *IBM Systems Journal*, vol. 33, no. 1, pp. 46-61.

Birmingham, A 1996, 'Users slam 'computer cowboys'', *ComputerWorld*, 23 Feb,18,29, p. 4.

Biro, M, Messnarz, R & Davison, A 2002, 'The impact of national cultural factors on the effectiveness of process improvement methods: the third dimension', *Software Quality Professional*, vol. 4, no. 4, pp. 34-41.

Black, T 1996, 'Quality rules added costs', *The Courier Mail*, 5 August, p. 4.

Blackburn, J, Scudder, G & Van Wassenhove, L 1996, 'Improving speed and productivity of software development: a global survey of software developers', *IEEE Transactions on Software Engineering*, vol. 22, no. 12, pp. 875-85.

Bogen, K 1996, 'The Effect of Questionnaire Length on Response Rates - A Review of the Literature', in *Proceedings of the Survey Research Methods Section*, American Statistical Association, pp. 1020-5.

Bollinger, TB & McGowan, C 1991, 'A critical look at software capability evaluations', *IEEE Software*, vol. 8, no. 4, pp. 25-41.

Bonate, PL 2000, *Analysis of pretest-posttest designs*, Chapman & Hall.

Borchers, G 2003, 'The software engineering impacts of cultural factors on multi-cultural software development teams', in *Proceedings of the 25th International Conference on Software Engineering*, IEEE Computer Society Washington, DC, USA, Portland, Oregon, pp. 540-5.

Boston Consulting Group 2001, *Towards a Software Coast: discussion paper*, Melbourne.

Brewer, J & Hunter, A 1989, *Multimethod Research: A Synthesis of Styles*, Sage Publications, Newbury Park, Calif.

Briand, L, El Emam, K & Morasca, S 1995, *On the Application of Measurement Theory in Software Engineering*, viewed 14 June 2004, <http://www2.umassd.edu/SWPI/ISERN/isern-95-04.pdf>.

Brilliant, SS & Knight, JC 1999, 'Empirical research in software engineering: a workshop', *ACM SIGSOFT Software Engineering Notes*, vol. 24, no. 3, pp. 44-52.

Brodman, JG & Johnson, DL 1994, 'What small businesses and small organisations say about the CMM', in *Proceedings of 16th International Conference on Software Engineering*, pp. 331-40.

---- 1995, 'Return on investment (ROI) from software process improvement as measured by US industry', *Software Process: Improvement and Practice*, no. Pilot issue, pp. 35-47.

---- 1997a, 'A software process improvement approach tailored for small organizations and small projects', in *Proceedings of the International Conference on Software Engineering (ICSE 19)*, IEEE Computer Soc, Boston, MA, pp. 661-2.

---- 1997b, *The LOGOS Tailored CMM for Small Business, Small Organizations and Small Projects*, LOGOS International Inc.

Brooks, F 1975, *The Mythical Man Month: Essays on Software Engineering*, Addison-Wesley, Reading, MA.

---- 1987, 'No silver bullet: essence and accidents of software engineering', *IEEE Computer*, vol. 20, no. 4, pp. 10-9.

BSA & PricewaterhouseCoopers 1998, *The Contribution of the Packaged Software Industry to the Australian Economy*, Business Software Alliance.

Bucci, G, Campanai, M & Cignoni, G 2001, 'Rapid assessment to solicit process improvement in small and medium-sized organisations', *Software Quality Professional*, vol. 4, no. 1, pp. 33-41.

Bullen, D 2003, 'Panel Session', paper presented to Australian Software and Systems Engineering Process Group (SEPG) Conference, Surfers Paradise, 25 Sept.

Burgess, A 1994, 'Studies try to quantify development trends', *IEEE Software*, vol. 11, no. 1, pp. 101-2.

Burstein, F & Gregor, S 1999, 'The systems development or engineering approach to research in information systems: an action research perspective', paper presented to 10th Australasian Conference on Information Systems, Wellington, NZ.

Butler, KL 1997, 'Process lessons learned while reaching level 4', *Crosstalk: The Journal of Defense Software Engineering*, May, pp. 1-6.

Carver, JC 2003, 'The impact of background and experience on software inspections', PhD thesis, University of Maryland.

Cater-Steel, AP 1999, 'SEA National Industry Improvement Program "Best Practice" Survey: Preliminary Findings', paper presented to Software Engineering Australia '99, Canberra, 1999.

---- 2000a, 'Software process evaluation: experience report from Down Under', in *Proceedings of European Conference on IT Evaluation*, Dublin, pp. 353-63.

---- 2000b, 'COTS developers lead best practice adoption', in *Proceedings of Australian Software Engineering Conference (ASWEC)*, IEEE, Canberra, pp. 23-30.

---- 2001, 'Process improvement in four small software companies', in *Proceedings of Australian Software Engineering Conference (ASWEC)*, Canberra, pp. 262-72.

---- 2002a, 'Process capability assessments in small development firms', in *Proceedings of IASTED 6th International Conference Software Engineering and Applications*, Cambridge, MA, pp. 737-42.

---- 2002b, 'Software process evaluation: experience report', *Electronic Journal of Information Systems Evaluation*, vol. 5, no. 2.

---- 2003, 'SPICE-based software process improvement: an empirical study of 22 small Australian firms', paper presented to Australian Software and Systems Engineering Process Group Conference, Surfers Paradise, 24-26 September.

---- 2004, 'Low-rigour, rapid software process assessments for small software development firms', in *Proceedings of Australian Software Engineering Conference (ASWEC)*, Melbourne, pp. 368-77.

Cater-Steel, AP, Toleman, MA & Rout, TP 2004, 'After the assessment: actions and reactions of 22 small Australian firms', in *Proceedings of 4th International SPICE Conference on Process Assessment and Improvement*, Lisbon, Portugal, pp. 54-63.

Cecez-Kecmanovic, D 1994, 'Engineering type information systems research: a discussion on its position and quality', paper presented to 5th Australian Conference on Information Systems, Melbourne.

Chiles, TH & Choi, TY 2000, 'Theorizing TQM: an Austrian and evolutionary economics interpretation', *Journal of Management Studies*, vol. 37, no. 2, pp. 185-212.

Chrissis, MB, Konrad, M & Shrum, S 2003, *CMMI: Guidelines for Process Integration and Product Improvement*, Addison-Wesley, Boston.

CMMI Product Team 2002, *Capability Maturity Model® Integration (CMMI), Version 1.1: CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing*, CMMI-SE/SW/IPPD/SS, V1.1, Carnegie Mellon University Software Engineering Institute, Pittsburgh PA.

Cnossen, C 1997, *Learning Paper 7 Secondary Research*, viewed 14 Mar 2001, <http://jura1.eee.rgu.ac.uk/modules/research/internal/resmeth/rmeth7_1.htm>.

Coakes, SJ & Steed, LG 1996, *SPSS for Windows: Analysis without Anguish*, Jacaranda Wiley.

Coleman, G 2002, '"Practice not process" - improving the capability of software startups', paper presented to XP 2002 Conference, Sardinia.

Cook, TD & Campbell, DT 1979, *Quasi-Experimentation: Design & Analysis Issues for Field Settings*, Rand McNally, USA.

Cornford, A & Smithson, S 1996, *Project Research in Information Systems*, Macmillan Press Ltd, Hampshire.

Cortada, JW 1997, *Best Practices in Information Technology*, Prentice Hall, Upper Saddle River, NJ.

Cox, K, Aurum, A & Jeffery, DR 2004, 'An experiment in inspecting the quality of use case descriptions', *Journal of Research and Practice in Information Technology*, vol. 36, no. 4, pp. 211-29.

CPA Australia 2002, *Small Business Survey Program: Perceptions of Risk*.

Cragg, PB 2002, 'Benchmarking information technology practices in small firms', *European Journal of Information Systems*, vol. 11, no. 4, pp. 267-82.

Crosby, P 1979, *Quality is Free*, McGraw-Hill, New York.

Culnan, MJ & Swanson, EB 1986, 'Research in management information systems 1980-1984: points of work and reference', *MIS Quarterly*, vol. 10, no. 3, pp. 289-301.

Curtis, B & Paulk, MC 1993, 'Creating a software process improvement program', *Information and Software Technology*, vol. 35, no. 6/7, pp. 381-6.

Curtis, B, Krasner, H & Iscoe, N 1988, 'A field study of the software design process for large systems', *Communications of the ACM*, vol. 31, no. 11, pp. 1268-87.

Curtis, B, Hefley, W & Miller, S 1995, *People Capability Maturity Model*, CMU/SEI-95-MM-02, Software Engineering Institute, Carnegie Mellon University, Pittsburgh.

Cusumano, MA, MacCormack, A, Kemerer, CF & Crandall, B 2003, 'Software development worldwide: the state of the practice', *IEEE Software*, vol. 20, no. 6, pp. 28-34.

Daft, RL 1998, *Essentials of Organization Theory and Design*, South-Western College Publishing, Cincinnati, Ohio.

Dale, BG, -Wu, PY, Zaire, M, Williams, ART & Van der Wiele, T 2001, 'Total quality management and theory: An exploratory study of contribution', *Total Quality Management*, vol. 12, no. 4, pp. 439-49.

Dale, D, Gilmour, D & Maltz, T 2002, 'Towards a Software Coast', *Software (SEA National)*, April, pp. 4-9.

Damanpour, F & Evan, WM 1984, 'Organizational innovation and performance: the problem of organizational lag', *Administrative Science Quarterly*, vol. 29, no. 3, pp. 392-409.

Dart, SA 1992, *The Past, Present and Future of Configuration Management*, CMU/SEI-92-TR-8; ESC-TR-92--8, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Davidson, P 2004, 'SMEs chase their share', *Information Age*, April/May, pp. 16-9.

Davis, CJ, Thompson, JB, Smith, P & Gillies, A 1992, *A Survey of Approaches to Software Quality within the United Kingdom*, University of Sunderland.

Davis, CJ, Gillies, AC, Smith, P & Thompson, JB 1993, 'Current practice in software quality and the impact of certification schemes', *Software Quality Journal*, vol. 2, pp. 145-61.

Davis, D & Cosenza, RM 1988, *Business Research for Decision Making*, 2 edn, PWS-Kent, Boston.

Davis, GB 1983, *Evolution of Information Systems as an academic discipline*, MISRRC-WP-83-07, Management Information Research Center, School of Management, University of Minnesota.

de Vaus, DA, Response Rate Calculation, e-mail to author,19/9/96.

Dean, JW & Bowen, DE 1994, 'Management theory and total quality: improving research and practice through theory development', *Academy of Management Review*, vol. 19, no. 3, pp. 392-418.

Debou, C & Kuntzmann-Combelles, A 2000, 'Linking software process improvement to business strategies: experiences from industry', *Software Process: Improvement and Practice*, vol. 5, pp. 55-65.

Deephouse, C, Goldenson, DR, Kellner, M & Mukhopadhyay, T 1995, 'The effects of software processes on meeting targets and quality', in *Proceedings of 28th Hawaii International Conference on System Sciences*, Western Periodicals, Hawaii, vol. 4, pp. 710-9.

Deming, WE 1986, *Out of the Crisis*, MIT Centre for Advanced Engineering Study, Cambridge, Mass.

Demirors, E, Demirors, O, Dikenelli, O & Keskin, B 2001, 'Process improvement towards ISO 9001 certification in a small software organisation', in RB Hunter & RH Thayer (eds), *Software Process Improvement*, IEEE.

Department of Defence 2004, *The Australian Defence Electronic Systems Sector Strategic Plan*, Commonwealth of Australia, Canberra.

Dependable Database Data 1998, *Australia on Disc*.

Diaz, M & Sligo, J 1997, 'How software process improvement helped Motorola', *IEEE Software*, vol. 14, no. 5, pp. 75-81.

Dillman, DA 1978, *Mail and Telephone Surveys: The Total Design Method*, Wiley-Interscience.

Dion, R 1992, 'Elements of a process-improvement program', *IEEE Software*, vol. 9, no. 4, pp. 83-5.

---- 1993, 'Process improvement and the corporate balance sheet', *IEEE Software*, vol. 10, no. 4, pp. 28-35.

Drehmer, D & Dekleva, S 1993, 'Measuring software engineering maturity: a Rasch calibration', paper presented to International Conference on Information Systems, Dec.

Dromey, RG & Rout, TP 1992, *Australia's Competitive Dependence on Software: Threats, Challenges and Opportunities*, SQI-92-01, ASQI, Griffith University, Brisbane.

Dunn, RH & Ullman, RS 1994, *TQM for Computer Software*, McGraw-Hill, New York.

Dutta, S & Van Wassenhove, LN 1997a, *An empirical study of adoption levels of software management practices within European firms*, 97/118/TM, INSEAD, Fontainebleau, France.

---- 1997b, *Management practices for software excellence: an empirical investigation*, 97/32/TM, INSEAD, Fontainebleau, France.

Dutta, S, Kulandaiswamy, S & Van Wassenhove, LN 1996, *Benchmarking European software management practices*, 96/45/TM, INSEAD, Fontainebleau, France.

Dutta, S, Van Wassenhove, LN & Kulandaiswamy, S 1998, 'Benchmarking European software management practices', *Communications of the ACM*, vol. 41, no. 6, pp. 77-86.

Dutta, S, Lee, MA & Van Wassenhove, LN 1998, 'Adoption levels of software management best practices in European organizations', paper presented to SPI'98 European Conference on Software Process Improvement: Improvement in Practice: Reviewing Experience, Previewing Future Trends, Monte Carlo.

Dutta, S, Lee, M & Van Wassenhove, L 1999, 'Software engineering in Europe: a study of best practices', *IEEE Software*, vol. 16, no. 3, pp. 82-90.

Dyba, T 2000, 'An instrument for measuring the key factors of success in software process improvement', *Empirical Software Engineering*, vol. 5, pp. 357-90.

---- 2002, 'Enabling software process improvement: an investigation of the importance of organisational issues', *Empirical Software Engineering*, vol. 7, pp. 387-90.

---- 2003, 'Factors of software process improvement success in small and large organizations: an empirical study in the Scandinavian context', paper presented to ESEC/FSE'03, Helsinki, Finland, Sept 1-5.

Easterby-Smith, M, Thorpe, R & Lowe, A 1991, *Management Research: An Introduction*, Sage, London.

Edwards, HM & Thompson, JB 2003, *Workshop on Sociotechnical Software Engineering Research*, viewed 20 June 2003, <http://www.cs.colorado.edu/serl/seworld/database/3449.html>.

Eisenhardt, KM 1988, 'Agency- and institutional-theory explanations: the case of retail sales commission', *Academy of Management Journal*, vol. 31, no. 3, pp. 488-511.

El Emam, K 1998, 'The internal consistency of the ISO/IEC 15504 software process capability scale', in *Proceedings of 5th International Symposium on Software Metrics*, pp. 72-81.

---- 2002, *Phase 2 Trials Final Report Volume 1*, viewed 8th November 2003, <http://www.cis.strath.ac.uk/research/SPICE/>.

El Emam, K & Madhavji, NH 1995, 'A field study of requirements engineering practices in information systems development', in *Proceedings of 2nd IEEE International Symposium on Requirements Engineering*, IEEE Press, York UK, pp. 68-80.

El Emam, K & Goldenson, DR 1998, *SPICE: An empiricist's perspective*, ISERN-98-03.

El Emam, K & Birk, A 2000a, 'Validating the ISO/IEC 15504 measure of software requirements analysis process capability', *IEEE Transactions on Software Engineering*, vol. 26, no. 6, pp. 541-66.

El Emam, K & Garro, I 2000, 'Estimating the extent of standards use: the case of ISO/IEC 15504', *Journal of Systems & Software*, vol. 53, no. 2, pp. 137-43.

El Emam, K & Birk, A 2000b, 'Validating the ISO/IEC 15504 measure of software development process capability', *Journal of Systems & Software*, vol. 51, no. 2, pp. 119-49.

El Emam, K, Fusaro, P & Smith, B 1999, 'Success factors and barriers for software process improvement', in R Messnarz & C Tully (eds), *Better Software Practice for Business Benefit*, IEEE CS Press, pp. 355-71.

El Emam, K, Goldenson, DR, McCurley, J & Herbsleb, J 1998, *Success or Failure? Modeling the Likelihood of Software Process Improvement*, ISERN-98-15, International Software Engineering Research Network (ISERN).

ESI *PIE's Repository*, European Software Institute, viewed 19 May 2003, <www.esi.es/VasieSearch/oneright.jsp>.

---- 1995, *1995 Software Best Practice Questionnaire Analysis of Results*, ESI-PIA-960464V5, European Software Institute, Bilboa.

---- 1996a, *Software Best Practice Questionnaire Comparative Analysis of 1995 and 1996*, ESI-PIA/96303, European Software Institute, Bilboa.

---- 1996b, *1996 Software Best Practice Questionnaire Analysis of Results*, ESI-1996-PIA/96281, European Software Institute, Bilboa.

---- 1997, *1997 Software Best Practice Questionnaire: Analysis of Results*, European Software Institute, Bilboa.

---- 1998, *Software Best Practice Questionnaire: Trend Analysis 1995-1997*, ESI-1998-ESI-98-TR-001, European Software Institute, Bilboa.

Evaristo, JR & Karahanna, E 1997, 'Is North American IS research different from European IS research?' *ACM SIGMIS Database*, vol. 28, no. 3, pp. 32-43.

Falconer, DJ & Mackay, DR 1999, 'The key to the mixed method dilemma', paper presented to Australasian Conference on Information Systems, Wellington.

Fayad, ME, Laitinen, M & Ward, RP 2000, 'Software engineering in the small', *Communications of the ACM*, vol. 43, no. 3, pp. 115-8.

Fenton, N 1993a, 'How effective are software engineering methods?' paper presented to 2nd International Conference on Achieving Quality in Software, Venice, Italy, 1993.

---- 1993b, 'How effective are software engineering methods?' *Journal of Systems & Software*, vol. 22, pp. 141-6.

Fenton, N, Pfleeger, SL & Glass, RL 1994, 'Science and substance: a challenge to software engineers', *IEEE Software*, vol. 11, no. 4, pp. 86-95.

Fichman, RG 1992, 'Information technology diffusion: a review of empirical research', in *Proceedings of 13th International Conference on Information Systems*, Dallas, pp. 195-206.

---- 1994, ‘The assimilation and diffusion of software process innovations’, in *Proceedings of International Conference on Information Systems - Doctorial Consortium*, Ernst & Young.

---- 1995, ‘The Assimilation and Diffusion of Software Process Innovations’, Ph D thesis, MIT.

Fichman, RG & Kemerer, CF 1993, ‘Adoption of software engineering process innovations: the case of object orientation’, *Sloan Management Review*, vol. 34, no. 2, pp. 7-22.

Finkelstein, A & Kramer, J 2000, ‘Software engineering: a roadmap’, in A Finkelstein (ed.), *The Future of Software Engineering*, ACM Press.

Fischer, SJ, Achterberg, J & Vinig, TG 1993, ‘Identifying different paradigms for managing information technology’, in *Proceedings of Conference on Computer Personnel Research*, ACM, St Louis, pp. 37-55.

Fitzgerald, B 1997, ‘The use of systems development methodologies in practice: a field study’, *Information Systems Journal*, vol. 7, pp. 201-12.

Fitzgerald, B & O’Kane, T 1999, ‘A Longitudinal Study of Software Process Improvement’, *IEEE Software*, pp. 37-45.

Fowler, FJ 1988, *Survey Research Methods*, Sage Publications, Newbury Park, CA.

Framework for the Future Mapping Working Group 2002, *An Overview of the Australian ICT Industry and Innovation Base*.

Frank, L, Sundqvist, S, Puumalainen, K & Taalikka, S 2001, ‘Cross-cultural comparison of innovators: empirical evidence from wireless services in Finland, Germany and Greece’, paper presented to ANZMAC Conference.

Fraser, P, Moultrie, J & Gregory, M 2002, ‘The use of maturity models/grids as a tool in assessing product development capability’, in *Proceedings of IEEE International Engineering Management Conference*, pp. 244-9.

French, WL & Bell, CH 1995, *Organization Development: Behavioral Science Interventions for Organization Improvement*, 5 edn, Prentice Hall International, Englewood Cliffs.

Gable, GG 1994, 'Integrating case study and survey research methods: an example in information systems', *European Journal of Information Systems*, vol. 3, no. 2, pp. 112-26.

Galin, D 2004, *Software Quality Assurance: from theory to implementation*, Pearson Education Ltd, Harlow Essex England.

Galletta, D & Lederer, A 1989, 'Some cautions on the measurement of user information satisfaction', *Decision Sciences*, vol. 20, pp. 419-38.

Galliers, RD 1991, 'Choosing appropriate information systems research approaches: a revised taxonomy', in H-E Nissen, HK Klein & R Hirschheim (eds), *Information Systems Research: contemporary approaches and emergent traditions*, Elsevier Science.

Gallivan, MJ 1997, 'Value in triangulation: a comparison of two approaches for combining qualitative and quantitative methods', in AS Lee, J Liebenau & J DeGross (eds), *Information Systems and Qualitative Research*, Chapman & Hall, pp. 417-43.

Gartner Group 1995, *MIS 3000 (database)*, Sydney, 1995.

Gasston, JL & Rout, TP 1994, 'Can the effectiveness of software processes be assessed?' *Software Quality Journal*, vol. 3, pp. 153-66.

Gay, LR & Diehl, PL 1992, *Research Methods for Business and Management*, Macmillan, New York.

Geck, B, Gloger, M, Jockusch, S, Lebsanft, K, Mehner, T, Paul, P, Paulish, F, Rheindt, M, Volker, A & Weber, N 1998, 'Software@Siemens: Best practices for the measurement and management of processes and architectures', paper presented to SPI'98 The European Conference on Software Process Improvement:

Improvement in Practice: Reviewing Experience, Previewing Future Trends, Monte Carlo.

Glass, RL 1996, 'The relationship between theory and practice in software engineering', *Communications of the ACM*, vol. 39, no. 11, pp. 11-3.

---- 2003a, *Facts and Fallacies of Software Engineering*, Addison-Wesley, Pearson, Boston.

---- 2003b, 'The state of the practice of software engineering', *IEEE Software*, vol. 20, no. 6, pp. 20-1.

Glass, RL, Vessey, I & Ramesh, V 2002, 'Research in software engineering: an analysis of the literature', *Journal of Information and Software Technology*, vol. 44, pp. 491-506.

Goldenson, DR & Herbsleb, JD 1995, *After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success*, CMU/SEI-95-TR-009, Software Engineering Institute, Carnegie Mellon University.

Goldenson, DR & Gibson, D 2003, *Demonstrating the impact and benefits of CMMI: an update and preliminary results*, CMU/SEI-2003-SR-009, Software Engineering Institute, Carnegie Mellon University, Pittsburgh.

Goldenson, DR, El Emam, K, Herbsleb, J & Deephouse, C 1997, *Empirical Studies of Software Process Assessment Methods*, ISERN-97-09, ISERN.

Goldsworthy, A 1997, *The Global Information Economy: The Way Ahead*, DIST, Commonwealth of Australia, Canberra.

Gome, A 1996a, 'Canberra undoes the quality push', *Business Review Weekly*, 8 July, pp. 29-30.

---- 1996b, 'Total Quality Madness', *Business Review Weekly*, 30 Sept, pp. 38-44.

Goode, S 2001, 'Organisational size metrics in IS research: a critical survey of the literature 1989-2000', in G Finnie, D Cecez-Kecmanovic & B Lo (eds), *Proceedings of 12th Australasian Conference on Information Systems*, Coffs Harbour, pp. 257-68.

Gottliebsen, R 1995, ‘Good sense ISO’, *Business Review Weekly*, 18 Sept 1995, p. 8.

Graettinger, C 2003, ‘I hired employees but people showed up’, paper presented to Australian Software and Systems Engineering Process Group Conference, Surfers Paradise, 24-26 September.

Graham, S & Mowery, DC 2003, ‘Intellectual property protection in the software industry’, paper presented to Science, Technology and Economic Policy Board: Patents in the Knowledge-based Economy.

Gray, L 1998, ‘Why coaches are needed in software process improvement’, *Crosstalk: The Journal of Defense Software Engineering*, Sept.

Graydon, AW, Nevalainen, R & Drouin, J 1998, ‘The reference model’, in K El Emam, J Drouin & W Melo (eds), *SPICE: the theory and practice of software process improvement and capability determination*, IEEE Computer Society, pp. 75-97.

Grogg, J 2003, *Conquering Empirical Research*, Mississippi State University Libraries, viewed 11 Dec 2004, <http://library.msstate.edu/ li/tutorial/empirical/>.

Grover, V & Teng, JTC 1992, ‘An examination of DBMS adoption and success in American organisations’, *Information and Management*, vol. 23, pp. 239-48.

Groves, L, Nickson, R, Reeve, G, Reeves, S & Utting, M 2000, ‘A survey of software development practices in the New Zealand software industry’, in *Proceedings of Australian Software Engineering Conference*, Canberra, pp. 189-201.

Grunbacher, P 1997, ‘A software assessment process for small software enterprises’, paper presented to EUROMICRO 97 New Frontiers of Information Technology, Proceedings of the 23rd EUROMICRO Conference, Budapest, 1-4 Sept.

Guba, EG & Lincoln, YS 1994, ‘Competing paradigms in qualitative research’, in NK Denzin & YS Lincoln (eds), *Handbook of Qualitative Research*, Sage Publications, Inc, Thousand Oaks, Calif., pp. 105-17.

Gutek, BA 1991, ‘Commentary (on Attewell and Rule’s paper)’, in K Kraemer (ed.), *The Information Systems Research Challenge: Survey Research Methods*, vol. 3, pp. 317-22.

Hackman, JR & Wageman, R 1995, ‘Total quality management: Empirical, conceptual, and practical issues’, *Administrative Science Quarterly*, vol. 40, no. 2, pp. 309-42.

Hailey, VA 1998, ‘A comparison of ISO9000 and the SPICE framework’, in K El Emam, J Drouin & W Melo (eds), *SPICE: the theory and practice of software process improvement and capability determination*, IEEE Computer Society, pp. 233-68.

Hall, T, Rainer, A & Baddoo, N 2002, ‘Implementing software process improvement: an empirical study’, *Software Process Improvement and Practice*, vol. 7, no. 1, pp. 3-15.

Hammer, M & Champy, J 1993, *Reengineering the Corporation*, HarperCollins, New York.

Hardy, CJ, Thompson, JB & Edwards, HM 1995, ‘The use, limitations and customization of structured systems development methods in the United Kingdom’, *Information and Software Technology*, vol. 37, no. 9, pp. 467-77.

Hayes, W & Zubrow, D 1995, *Moving on up: data and experience doing CMM-based process improvement*, Software Engineering Institute, Carnegie Mellon University, viewed 10 December 2004, <http://www.sei.cmu.edu/pub/documents/95.reports/pdf/tr008.95.pdf>.

Henninger, A, Sieber, A & Dilger, W 2001, *Varying Cultures – A Challenge For Software Engineering?*, Chemnitzer Informatik-Berichte, TU Chemnitz, viewed 20 June 2004, <http://www.tu-chemnitz.de/phil/NeueMedien/SWE/Download/CIB_2001_2.pdf>.

Henry, J, Henry, S, Kafura, D & Matheson, L 1994, 'Improving software maintenance at Martin Marietta', *IEEE Software*, vol. 11, no. 4, pp. 67-75.

Herbsleb, J & Goldenson, DR 1996, 'A systematic survey of CMM experience and results', in *Proceedings of the 18th International Conference on Software Engineering*, IEEE Computer Society Press, Berlin, pp. 323-30.

Herbsleb, J, Zubrow, D, Siegal, J, Rozum, J & Carleton, A 1994, 'Software process improvement: state of payoff', *American Programmer*, vol. 7, no. 9, pp. 2-12.

Hersh, A 1993, 'Where's the return on process improvement?' *IEEE Software*, vol. 10, no. 4, p. 12.

Hilvert, J 1996, 'A ticket to the dance: is the Government-endorsed supplier scheme worth the price of admission?' *Australian Reseller News*, vol. 1, no. 2, pp. 29,31.

Ho, N & McGuire, E 1998, 'A case study on the software culture at a rank startup: an argument for adopting a mature, corporate, and process-driven mentality', paper presented to AIS 1998 Americas Conference.

Hofstede, G *Hofstede's Dimension of Culture Scales*, viewed 10 May 2004, <http://spectrum.troyst.edu/~vorism/hof_ban.htm>.

Holmes, S & Gibson, B 2001, *Definition of Small Business*, University of Newcastle, Newcastle.

Horvat, RV, Rozman, I & Gyorkos, J 2000, 'Managing the complexity of SPI in small companies', *Software Process: Improvement and Practice*, vol. 5, pp. 45-54.

Houghton, J 2003, *Australian ICT Industries Update 2002 Executive Summary*, Centre for Strategic Economic Studies.

Howarth, B 2004a, 'Software shut-out', *Business Review Weekly*, vol. 26, no. 1, pp. 29-31.

---- 2004b, 'Slick software', *MIS Australia*, vol. 13, no. 2, pp. 72-5.

Huck, SW, Cormier, WH & Bounds, WG 1974, *Reading Statistics and Research*, Harper & Row, New York.

Humphrey, WS 1990, *Managing the Software Process*, Addison-Wesley, Reading, Mass.

Humphrey, WS, Kitson, DH & Kasse, TC 1989, *The State of Software Engineering Practice: A Preliminary Report*, CMU/SEI-89-TR-1, Software Engineering Institute, Carnegie Mellon University, Pittsburgh.

Humphrey, WS, Snyder, TR & Willis, RR 1991, 'Software process improvement at Hughes Aircraft', *IEEE Software*, vol. 8, no. 4, pp. 11-23.

Humphrey, WS, Kitson, DH & Gale, J 1991, 'A comparison of U.S. and Japanese software process maturity', paper presented to 13th International Conference on Software Engineering, Austin, Texas, 1991.

Hunter, R 1998, 'SPICE trials assessment profile', *IEEE Software Process Newsletter*, vol. 12, pp. 12-8.

---- 1999, 'Software process assessment: difficulties and benefits', paper presented to 6th European Conference on Software Quality.

Hunter, R & Jung, H 2000, 'The regional factor in software process assessment', paper presented to 2nd World Congress on Software Quality.

Hwang, J & Jung, H 2003, 'Validating the process capability measures of project management process in ISO/IEC PDTR 15504', paper presented to SPICE Conference, Noordwijk, Netherlands.

Ibanez, M & Reo, D 1998, 'Aligning software process improvement with business goals: ESI Balanced IT Scorecard', paper presented to European Software Engineering Process Group Conference, London.

Ibbs, CW & Kwak, YH 2000, 'Assessing project management maturity', *Project Management Journal*, vol. 31, no. 1, pp. 32-43.

IBISWorld 2004, *L7834 - Computer Consultancy Services in Australia*, viewed 15 Oct 2004, <http://www.ibisworld.com.au/alliance/buschan/industry.asp?industry_id=558>.

IDG Staff 2003, 'Government billions on offer, but not much for SMEs', *Information Age*, April/May, pp. 17-8.

IEEE 1990, *IEEE Standard Glossary of Software Engineering Terminology*, Std 610.12-1990, IEEE, Piscataway NJ.

IIB 1997, *Queensland IT&T Industry Survey*, Information Industries Bureau, Brisbane.

---- 2000, *Queensland ICT Industry Profile 1999/2000*, Information Industries Bureau, Department of Innovation and Information Economy, Queensland Government, Brisbane.

Iivari, J 1991, 'A paradigmatic analysis of contemporary schools of IS development', *European Journal of Information Systems*, vol. 1, no. 4, pp. 249-72.

Iivari, J, Hirschheim, R & Klein, HK 1998, 'A paradigmatic analysis contrasting information systems development approaches and methodologies', *Information Systems Research*, vol. 8, no. 2, pp. 164-93.

Inwood, C 1993, 'Developers still lagging in ISO preparation', *Computing Canada*, vol. 19, no. 17, p. 19.

ISBSG 1998, *The Benchmark Release 5*, International Software Benchmarking Standards Group, viewed 2 Sept 1999, <http://www.isbsg.org.au /bmarkr5.htm>.

ISO/IEC 2002, *Information technology - Software life cycle processes: Amendment 1*, ISO/IEC 12207 AMD 1.

ISO/IEC JTC1/SC7 N872 1991, *Proposal for a Study Period on Process Management*.

ISO/IEC JTC1/SC7 N944R 1992, *The Need and Requirements for a Software Process Assessment Standard, Study Report, Issue 2.0*.

ISO/IEC PDTR 15504 1996, *Information Technology - Software Process Assessment: Parts 1-9*.

ISO/IEC TR 15504-2 1998, *Information technology -Software process assessment - Part 2: A reference model for processes and process capability*, ISO/IEC TR 15504-2:1998(E).

ISO/IEC TR 15504-5 1998, *Information technology -Software process assessment - Part 5: An assessment model and indicator guidance*, ISO/IEC TR 15504-5:1998(E).

ISO/IEC TR 15504-6 1998, *Information technology -Software process assessment - Part 6: Guide to competency of assessors*, ISO/IEC TR 15504-6:1998(E).

ISO/IEC TR 15504-9 1998, *Information technology -Software process assessment - Part 9: Vocabulary*, ISO/IEC TR 15504-9:1998(E).

ISO/IEC TR 15504 1998, *Information Technology -Software process assessment, Parts 1-9*, ISO/IEC TR 15504:1998(E).

Iversen, J, Johansen, J, Nielsen, PA & Pries-Heje, J 1998, 'Combining quantitative and qualitative assessment methods in software process improvement', in Baets (ed.), *Proceedings of 6th European Conference on Information Systems (ECIS)*, Aix-en-Provence, France, pp. 451-66.

Jalote, P 2001, 'The success of the SPI efforts in India', *Software Quality Professional*, vol. 3, no. 2.

James, A 2000, 'Can Australia improve its software processes?' *Software (SEA National)*, Summer, pp. 6-9.

---- 2001, 'Can Australia improve its software processes?' *Crosstalk: The Journal of Defense Software Engineering*, August.

James, A & Hont, J 1998, *Stocktake of Australia's Information Industries*, Department of Industry, Science and Tourism, Canberra.

Jarvinen, J 2000, 'Measurement based continuous assessment of software engineering processes', PhD thesis, Oulu.

Jeffery, DR 1993, 'A view on the use of three research philosophies to address empirically determined weaknesses of the software engineering process', in HD Rombach, V Basili & RW Selby (eds), *Experimental Software Engineering Issues: critical assessment and future directions: Proceedings International Workshop 1992*, Springer-Verlag, Dagstuhl Castle, Germany, pp. 111-5.

Jeffery, DR & Zucker, B 1997, *The state of practice in software metrics*, Technical Report 97/1, School of Information Systems, UNSW, Sydney.

Jeffery, DR & Votta, LG 1999, 'Guest Editor's special section introduction: empirical software engineering', *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 435-7.

Jenner, MG 1995, *Software Quality Management and ISO 9001*, John Wiley & Sons, New York.

Jick, TD 1979, 'Mixing qualitative and quantitative methods: triangulation in action', *Administrative Science Quarterly*, vol. 24, pp. 602-9.

Johns, BL, Dunlop, WC & Sheehan, WJ 1989, *Small Business in Australia: Problems and Prospects*, 3rd edn, Allen and Unwin, Sydney.

Johnson, A 1994, 'Software process improvement experience in the DP/MIS function: experience report', paper presented to Proceedings of 16th International Conference on Software Engineering (ICSE), May 1994.

Johnson, DL & Brodman, JG 1992, 'Software process rigors yield stress, efficiency', *Signal*, pp. 55-7.

Jones, C 1994, *Assessment and Control of Software Risks*, Prentice Hall.

---- 1996, 'The pragmatics of software process improvements', *Software Process Newsletter of the Software Engineering Technical Council*, vol. 5, no. Winter, pp. 1-4.

---- 2003, ‘Variations in software development practices’, *IEEE Software*, pp. 22-7.

Jung, H & Hunter, R 2001, ‘The relationship between ISO/IEC 15504 process capability levels, ISO 9001 certification and organisation size: an empirical study’, *Journal of Systems & Software*, vol. 59, no. 1, pp. 43-55.

Jung, H, Hunter, R, Goldenson, DR & El Emam, K 2001, ‘Findings from phase 2 of the SPICE trials’, *Software Process: Improvement and Practice*, vol. 6, no. 4, pp. 205-42.

Juran, JM 1988, *Juran on Planning for Quality*, Macmillan, New York NY.

Kalafatis, SP & Tsogas, MH 1994, ‘Impact of the inclusion of an article as an incentive in industrial mail surveys’, *Industrial Marketing Management*, vol. 23, no. 2, pp. 137-43.

Kaltio, T & Kinnula, A 2000, ‘Deploying the defined SW process’, *Software Process Improvement and Practice*, vol. 5, no. 1, pp. 65-83.

Kan, SH 2003, *Metrics and Models in Software Quality Engineering*, 2 edn, Addison-Wesley, Boston.

Kaplan, B & Duchon, D 1988, ‘Combining qualitative and quantitative methods in information systems research: a case study’, *MISQ*, vol. 12, no. 4, pp. 571-86.

Kasse, T & McQuaid, PA 1998, ‘Entry strategies into the process improvement initiative’, *Software Process: Improvement and Practice*, vol. 4, pp. 73-88.

Katz, ML & Shapiro, C 1985, ‘Network externalities, competition, and compatibility’, *The American Economic Review*, vol. 75, no. 3, pp. 424-40.

Kautz, K 1998a, ‘Even in very small software enterprises metrics can make sense!’ paper presented to SPI’98 The European Conference on Software Process Improvement: Improvement in Practice: Reviewing Experience, Previewing Future Trends, Monte Carlo, Dec.

---- 1998b, ‘Software process improvement in very small enterprises: does it pay off?’ *Software Process: Improvement and Practice*, vol. 4, pp. 209-26.

Kautz, K, Hansen, HW & Thaysen, K 2001, 'Applying and adjusting a software process improvement model in practice: the use of the IDEAL model in a small software enterprise', in RB Hunter & RH Thayer (eds), *Software Process Improvement*, IEEE, pp. 463-70.

Kelly, DP & Culleton, B 1999, 'Process improvement for small organisations', *IEEE Computer*, vol. 32, no. 10.

Khurana, M, He, Z, Court, I, Ross, M, Staples, G & Wilson, D 1996, 'Software quality practices - an empirical study', *Software Quality Journal*, no. 5, pp. 75-85.

King, WR 1996, 'Achieving global strategic advantage', *Information Systems Management*, vol. 13, no. 4, pp. 57-60.

Kitchenham, B 1996, 'Evaluating software engineering methods and tools. Part 1: The evaluation context and evaluation methods', *ACM SIGSOFT Software Engineering Notes*, vol. 21, no. 1, pp. 11--4.

Kitchenham, B, Pfleeger, SL, Pickard, LM, Jones, PW, Hoaglin, DC, El Emam, K & Rosenberg, J 2002, 'Preliminary guidelines for empirical research in software engineering', *IEEE Transactions on Software Engineering*, vol. 28, no. 8, pp. 721-34.

Kitson, DH 1996, 'Relating the SPICE framework and SEI approach to Software Process Assessment', *Software Quality Journal*, vol. 5, pp. 145-56.

Kitson, DH & Masters, S 1993, 'An analysis of SEI software process assessment results: 1987-1991', paper presented to Proceedings of 15th International Conference on Software Engineering (ICSE), May.

Kling, R, Kraemer, K, Allen, J, Bakos, Y, Gurbaxani, V & King, JL 1992, 'Information systems in manufacturing coordination: economic and social perspectives', paper presented to International Conference on Information Systems (ICIS), Dallas, Texas.

Komiyama, T, Sunazuka, T & Koyama, S 1998, 'Software process assessment and improvement in NEC: current status and future direction', paper presented to

SPI'98 The European Conference on Software Process Improvement: Improvement in Practice: Reviewing Experience, Previewing Future Trends, Monte Carlo.

Krasner, H 1997, *Accumulating the Body of Evidence for the Payoff of Software Process Improvement - 1997*, viewed 14 Mar 2001, <http://www.utexas.edu/coe/sqi/archive/krasner/spi.pdf>.

Krutchen, P 2004, *Analyzing intercultural factors affecting global software development – a position paper*, University of British Columbia, viewed 20 June 2004, <http://gsd2004.uvic.ca/camera/kruchten.pdf>.

Kulpa, M 1998, 'Ten things your mother never told you about the Capability Maturity Model', *Crosstalk: The Journal of Defense Software Engineering*, Sept.

Laitinen, M, Fayad, ME & Ward, RP 2000, 'Software engineering in the small', *IEEE Software*, vol. 17, no. 5, pp. 75-7.

Larsen, EA & Kautz, K 1997, 'Quality assurance and software process improvement in Norway', *Software Process Improvement and Practice*, vol. 3, no. 2, pp. 71-86.

Lau, F 1997, 'A review on the use of action research in information systems studies', in LA Liebenau & J DeGross (eds), *Information Systems and Qualitative Research*, Chapman & Hall, London, pp. 31-68.

Laudon, KC & Laudon, JP 2004, *Management Information Systems: Managing the Digital Firm*, Prentice Hall, Upper Saddle River, New Jersey.

Lawlis, PK, Flowe, RM & Thordahl, JB 1995, 'A correlational study of the CMM and software development performance', *Crosstalk: The Journal of Defense Software Engineering*, Sept.

Lawthers, I 1999, *Spire European Analysis Report: Esprit - ESSI Project 23973*, D2.3, Centre for Software Engineering, Dublin City University, Dublin.

Lee, AS 1991, 'Integrating positivist and interpretive approaches to organizational research', *Organization Science*, vol. 2, pp. 342-65.

Leonard-Barton, D 1987, 'Implementing structured software methodologies: a case of innovation in process technology', *Interfaces*, vol. 17, no. 3, pp. 6-17.

Liackman, F 1994, 'Quality accreditation - It costs to be without it', *Process Engineering*, vol. 22, no. 3, pp. 14-6.

Likert, R 1967, *The Human Organization: Its Management and Value*, McGraw-Hill, New York.

Loane, S 2003, 'Exporting small software firms: a cross-national comparison of the export success factors of software firms', *Software (SEA National)*, pp. 49-52.

Lowry, GR, Morgan, GW & FitzGerald, DG 1996, 'Identifying excellence in leading Australian-owned information technology firms: five emerging themes', paper presented to 7th Australian Conference on Information Systems.

Ludewig, J 2001, 'Software engineering in the year 2000 minus and plus ten', in R Wilhelm (ed.), *Informatics: 10 years back, 10 years ahead.*, Springer-Verlag, Berlin, Heidelberg, pp. 102 - 11.

Maani, KE, Putterill, MS & Sluti, DG 1994, 'Empirical analysis of quality improvement in manufacturing', *International Journal of Quality & Reliability Management*, vol. 11, no. 7, pp. 19-37.

Mahajan, V & Peterson, RA 1985, *Models for Innovation Diffusion*, Sage University Paper, Beverly Hills.

Mahoney, D, Trigg, M, Griffin, R & Pustay, M 2001, *International Business: A Managerial Perspective*, Pearson Education Australia.

Marshall, D & Hofmann, G 2001, 'Software standards in Defence contracting', *Software (SEA National)*, pp. 24-5.

McBride, T 2004, 'Standards need more rigour', *Information Age*, Oct/Nov, pp. 65-6.

McCaffery, F, Wilkie, FG, McFall, D & Lester, N 2004, 'Northern Ireland software industry survey', in *Proceedings of 4th International SPICE Conference on Process Assessment and Improvement*, Lisbon, Portugal, pp. 159-61.

McConnell, S 2002, ‘Closing the gap’, *IEEE Software*, vol. 19, no. 1, pp. 3-5.

---- 2003, *Professional Software Development: Shorter Schedules, Higher Quality Projects, More Successful Projects, Enhanced Careers*, Addison-Wesley, Boston.

McGibbon, T 1999, *A Business Case for Software Process Improvement Revised: an Updated DACS State-of-the-Art Report*, Department of Defense, Data & Analysis Centre for Software.

McGrath, JE, Martin, J & Kulka, RA 1982, *Judgment Calls in Research*, Sage, Beverly Hills.

McKerlie Consulting 1996, *Needs Analysis for Enhancement for Software Development Capability within Australian Industry*, DIST.

McKinsey & Co. 2002, *Australia: Winning in the Global ICT Industry*, Dept of Communications, Information Technology and the Arts.

Meehan, B & Richardson, I 2002, ‘Identification of software process knowledge management’, *Software Process: Improvement and Practice*, vol. 7, no. 2, pp. 47-55.

Milgrom, PR 1988, ‘Employment contracts, influence activities, and efficient organization design’, *Journal of Political Economy*, vol. 96, no. 1, p. 42.

Mili, H, Mili, F & Mili, A 1995, ‘Reusing software: issues and research directions’, *IEEE Transactions on Software Engineering*, vol. 21, no. 6, pp. 528-61.

Miller, J 2004, ‘Statistical significance testing - a panacea for software technology experiments?’ *The Journal of Systems and Software*, vol. 73, pp. 183-92.

Mingers, J 2001, ‘Combining IS research methods: towards a pluralist methodology’, *Information Systems Research*, vol. 12, no. 3, pp. 240-59.

Mintzberg, H 1983, *Structure in fives : designing effective organizations*, Prentice-Hall, Englewood Cliffs, N.J. USA.

Mohnot, N 1995, ‘Quality: made in India’, *American Programmer*, vol. 8, no. 10, pp. 12-8.

Moore, GA 1999, *Crossing the chasm: marketing and selling high-tech products to mainstream customers*, HarperBusiness, New York.

Moore, GC 1989, ‘An examination of the implementation of information technology for end-users: a diffusion of innovations perspective’, PhD thesis, University of British Columbia.

Moore, GC & Benbasat, I 1991, ‘Development of an instrument to measure the perceptions of adopting an information technology innovation’, *Information Systems Research*, vol. 2, no. 3, pp. 192-222.

Morrison, J & George, JF 1995, ‘Exploring the software engineering component in MIS research’, *Communications of the ACM*, vol. 37, no. 7, pp. 80-91.

Mustonen-Ollila, E & Lyytinen, K 2003, ‘Why organisations adopt information system process innovations: a longitudinal study using Diffusion of Innovation theory’, *Information Systems Journal*, vol. 13, pp. 275-97.

Myers, MD 1997, *Qualitative research in information systems*, MISQ Discovery, <www.misq.org/misqd961/isworld/>.

Natwick, G, Draper, G & Bearden, L 1999, ‘Software mini-assessments: process and practice’, *Crosstalk: The Journal of Defense Software Engineering*, Oct, pp. 10-4.

Neill, CJ & Laplante, PA 2003, ‘Requirements engineering: the state of the practice’, *IEEE Software*, vol. 20, no. 6, pp. 40-5.

Ng, SP, Murnane, T, Reed, K, Grant, D & Chen, TY 2004, ‘A preliminary survey on software testing practices in Australia’, in *Proceedings of Australian Software Engineering Conference*, Melbourne, pp. 116-25.

Niazi, M, Wilson, D & Zowghi, D 2003, ‘Critical success factors and critical barriers for software process improvement: an analysis of literature’, paper presented to

14th Australasian Conference on Information Systems, Perth, 26-28 November.

Nolan, AJ 1999, ‘Learning from success’, *IEEE Software*, vol. 16, no. 1, pp. 97-105.

Nolan, RL 1973, ‘Managing the computer resource: a stage hypothesis’, *Communications of the ACM*, vol. 16, pp. 339-405.

NZ Ministry of Economic Development 2001, *SMEs in New Zealand: Structure and Dynamics*, viewed 29 May 2004, <http://www.med.govt.nz/irdev/ind_dev/smes/2001/smes2001-14.html>.

O’Regan, N & Ghobadian, A 2004, ‘Testing the homogeneity of SMEs: the impact of size on managerial and organisational processes’, *European Business Review*, vol. 16, no. 1, pp. 64-77.

Orlikowski, WJ & Baroudi, JJ 1991, ‘Studying information technology in organizations: research approaches and assumptions’, *Information Systems Research*, vol. 2, no. 1, pp. 1-28.

Paulish, DJ & Carleton, AD 1994, ‘Case studies of software-process-improvement measurement’, *Computer*, vol. 27, no. 9, pp. 50-7.

Paulk, MC 1995, ‘How ISO 9000 compares with the CMM’, *IEEE Software*, vol. 12, no. 1, pp. 74-83.

Paulk, MC, Humphrey, WS & Pandelios, GJ 1992, ‘Software process assessments: issues and lessons learned’, in *Proceedings of ISQE92*, pp. 1-22.

Paulk, MC, Goldenson, DR & White, DM 2000, *The 1999 Survey of High Maturity Organizations*, CMU/SEI-2000-SR-002, Software Engineering Institute, Carnegie Mellon University, Pittsburgh.

Paulk, MC, Weber, C, Curtis, B & Chrissis, MB (eds) 1995, *Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, Reading, MA.

Paxson, CM 1992, 'Follow-up mail surveys', *Industrial Marketing Management*, vol. 21, no. 3, pp. 195-201.

Perry, DE, Porter, AA & Votta, LG 2000, 'Empirical studies of software engineering: a roadmap', in *Proceedings of the Conference on The Future of Software Engineering*, ACM Press, pp. 345-55.

Perry, JL & Danzinger, JN 1980, 'The adoptability of innovations: an empirical assessment of computer applications in local governments', *Administration and Society*, vol. 11, no. 4, pp. 461-92.

Pfleeger, SL 1994, 'Design and analysis in software engineering. Part 1: The language of case studies and formal experiments', *ACM SIGSOFT Software Engineering Notes*, vol. 19, no. 4, pp. 16-20.

Pfleeger, SL, Fenton, N & Page, S 1994, 'Evaluating software engineering standards', *IEEE Computer*, vol. 27, no. 9, pp. 71-9.

Pfleeger, SL, Jeffery, DR, Curtis, B & Kitchenham, B 1997, 'Status report on software management', *IEEE Software*, vol. 14, no. 2, pp. 33-43.

Phillips, M 2003, 'CMMI appraisal tutorial', paper presented to Australian Software Engineering Process Group (SEPG), Surfers Paradise.

Pitterman, W 2000, 'Telcordia Technologies: the journey to high maturity', *IEEE Software*, vol. 18, no. 4, pp. 89-96.

Powell, TC 1995, 'Total quality management as competitive advantage: a review and empirical study', *Strategic Management Journal*, vol. 16, pp. 15-37.

Prescott, MB & Conger, SA 1995, 'Information technology innovations: a classification by IT locus of impact and research approach', *Data Base Advances*, vol. 26, no. 2&3, pp. 20-41.

Pressman, RS 2003, *Software Engineering Resources*, R.S. Pressman & Associates, viewed 27 July 2004, <http://www.rspa.com/spi/process-agile.html#general.>.

Pringle, L 2001, ‘Size does matter: improvement for SMEs’, *Software (SEA National)*, no. Summer, pp. 4-7.

Qld Dept of State Development 2004, *ICT in the Smart State*, Brisbane.

QPS 2003, *Wageline: Queensland Public Service Award - State*, Queensland Government, viewed 22 August 2004, <http://www.wageline.qld.gov.au/awardsacts/showDoc.jsp?Awards/Q0110/5.4+Technical+stream>.

Quann, ES 1997, ‘“My boss needs to hear this”: how management can support software process improvement’, *Crosstalk: The Journal of Defense Software Engineering*, May.

Radice, RA, Harding, JT, Munnis, PE & Phillips, RW 1985, ‘A programming process study’, *IBM Systems Journal*, vol. 24, no. 2, pp. 79-90.

Raffa, M & Zollo, G 1993, ‘The role of professionals in Italian small software firms’, in *Proceedings of 2nd International Conference on Achieving Quality in Software,*, Venice, Italy, pp. 335-48.

Raffa, M, Zollo, G & Caponi, R 1996, ‘The short but interesting life of small software firms’, in S Bologna & G Bucci (eds), *Achieving Quality in Software*, Chapman and Hall,Great Britain, Florence, pp. 103-17.

Rainer, A & Hall, T 2001, ‘An Analysis of some ‘core studies’ of SPI’, *Software Process: Improvement and Practice*, vol. 6, no. 4, pp. 169-87.

---- 2002, ‘Key success factors for implementing software process improvement: a maturity-based analysis’, *The Journal of Systems and Software*, vol. 62, pp. 71-84.

---- 2003, ‘A quantitative and qualitative analysis of factors affecting software processes’, *The Journal of Systems and Software*, vol. 66, pp. 7-21.

Reeb, J & Henderson, R 1997, ‘Experiences in establishing mentor relationships for software process improvement efforts’, *Crosstalk: The Journal of Defense Software Engineering*, Nov.

Repenning, NP 2000, ‘Drive out fear (unless you can drive it in): the role of agency and job security in process improvement’, *Management Science*, vol. 46, no. 11, p. 1385.

Revenaugh, DL 1992, ‘Information systems architecture implementation: the impact of corporate culture and the strategic importance of information systems’, Arizona State University.

Richardson, I 1999, ‘Improving the software process in small indigenous software development companies using a model based on quality function deployment’, PhD thesis, University of Limerick.

---- 2001, ‘Software process matrix: a small company SPI model’, *Software Process: Improvement and Practice*, vol. 6, pp. 157-65.

---- 2002, ‘Software process improvements in very small companies’, *Software Quality Professional*, vol. 4, no. 2, pp. 14-22.

Richardson, I & Ryan, K 2001, ‘Software Process Improvements in a Very Small Company’, *Software Quality Professional*, vol. 3, no. 2, pp. 23-35.

Ridley, G & Keen, C 1999, ‘Epistemologies for information systems research’, in *Proceedings of the 9th Australian Conference on Information Systems*, UNSW, Sydney, pp. 520-32.

Riemenschneider, CK, Hardgrave, BC & Davis, FD 2002, ‘Explaining software developer acceptance of methodologies: a comparison of five theoretical models’, *IEEE Transactions on Software Engineering*, vol. 28, no. 12, pp. 1135-45.

Rifkin, S 2001, ‘Why software process innovations are not adopted’, *IEEE Software*, vol. 18, no. 4, pp. 110-2.

Rigby, C & Trantom, J 2004, *SMEs: A World View*, viewed 29 May 2004, <http://www.prospects.ac.uk/cms/ShowPage/Home_page/Explore_job_sectors/SMEs/A_world_view/p!eklmibL>.

Robey, D & Zmud, RW 1992, 'Research on the organization of end-user computing: theoretical perspectives from organization science', in *Information Technology and People*, Northwind, USA, pp. 11-27.

Rockart, JF 1979, 'Chief executives define their own data needs', *Harvard Business Review*, vol. 57, no. 2, pp. 81-93.

Rogers, EM 1995, *Diffusion of Innovations*, Free Press, New York.

Ronen, S & Shenkar, O 1985, 'Clustering countries on attitudinal dimensions: a review and synthesis', *Academy of Management Review*, vol. 10, no. 3, pp. 435-54.

Rout, TP 1992, 'Software Quality and Maturity of the Software Process', *The Source*, no. June, p. 6.

---- 1996, 'The SPICE Project - Past, Present and Future', paper presented to Software Process '96 Conference, Brighton UK, 1996.

---- 1998, 'Evaluating improvement strategies using software process assessment', paper presented to SPI'98 The European Conference on Software Process Improvement: Improvement in Practice: Reviewing Experience, Previewing Future Trends, Monte Carlo, 1998.

---- 2003, 'ISO/IEC 15504 - Evolution to an international standard', *Software Process Improvement and Practice*, vol. 8, no. 1, pp. 27-40.

Rout, TP & Griffith University 1998, *SPICE Assessor Training Course: Presenting the results*.

Rout, TP & Simms, PG 1998, 'Introduction to the SPICE documents and architecture', in K El Emam, J Drouin & W Melo (eds), *SPICE: the theory and practice of software process improvement and capability determination*, IEEE Computer Society, pp. 31-74.

Rout, TP, Neilson, MP & Gasston, JL 1994, 'Experiences with the use of different approaches to Software Process Assessment', in M Ross, CA Brebbia, G Staples & J Stapleton (eds), *Software Quality Management II Vol 1: Managing*

Quality Systems, Computational Mechanics Publications, Southampton Boston, pp. 767-81.

Rout, TP, Tuffley, A & Hodgen, B 1998, 'Process capability in the Australian software industry: results from the SPICE trials', in *Proceedings of Australian Software Engineering Conference (ASWEC)*, pp. 12-21.

Rout, TP, Hodgen, B & Tuffley, A 1999a, *Process Model for the Rapid Assessment of Process Capability*, SQI-GU, Brisbane.

---- 1999b, *Method for the Rapid Assessment of Process Capability*, SQI-GU, Brisbane.

Rout, TP, Tuffley, A & Cahill, B 2001, *Report for DMO - CMMI Evaluation: Capability Maturity Model Integration Mapping to ISO/IEC 15504-2:1998*, SQI, Brisbane.

Rout, TP, Tuffley, A, Cahill, B & Hodgen, B 2000, 'The rapid assessment of software process capability', paper presented to SPICE 2000 - 1st International Conference on Software Process Improvement and Capability dEtermination, Limerick.

Rubin, DB, Stern, HS & Vehovar, V 1995, 'Handling 'don't know' survey responses', *Journal of the American Statistics Association*, vol. 90, no. 431, pp. 822-8.

Rubin, H 1990, 'How to become a software engineering "Big Foot"', *American Programmer*, vol. 3, no. 1, pp. 22-32.

---- 1991, 'Software process maturity', *American Programmer*, vol. 4, no. 1, pp. 20-7.

Saiedian, H & Chennupati, K 1999, 'Towards an evaluative framework for software process improvement models', *The Journal of Systems and Software*, vol. 47, pp. 139-48.

Saran, C 2001, *The road to software excellence for SMEs*, ComputerWeekly.com, viewed 6 October 2003, <www.cw360.com/articles/article.asp?1iArticleID=104469&1iFlavourID=1>.

Sawyer, S 2001, ‘Analysis by long walk: some approaches to the synthesis of multiple sources of evidence’, in EM Trauth (ed.), *Qualitative Research in IS: Issues and Trends*, Idea Group Publishing, Hershey, PA, pp. 163-90.

Scacchi, W 1993, ‘Qualitative techniques and tools for measuring, analyzing, and simulating software processes’, in HD Rombach, V Basili & RW Selby (eds), *Experimental Software Engineering Issues: critical assessment and future directions*, Proceedings International Workshop 1992, Dagstuhl Castle, Germany edn, Springer-Verlag, pp. 27-9.

Schwalbe, K 2003, *Information technology Project Management*, 3 edn, Thomson Course Technology.

Scott, L, Jeffery, R, Carvalho, L, D’Ambra, J & Rutherford, P 2001, ‘Practical software process improvement - the IMPACT project’, paper presented to Australian Software Engineering Conference, Canberra, 27-28 Aug.

SEA 2003, *The Whitehorse Report: Software Industry Research Report*.

SEA National 2001, *Multinational Enterprise Benchmark Survey: Executive Summary*.

---- 2002, ‘Software Industry Report: Executive Summary’.

---- 2004, ‘Media Release: SEA Launches free Software Process Spot-Check initiative’, vol. 2004, no. 16 Oct.

Seaman, CB 1999, ‘Qualitative methods in empirical studies of software engineering’, *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 557-72.

SEAQ 1997, *Software Engineering Network Business Plan - Technical Support Centres Program*.

Seidman, SB, Bagert, D, Frailey, D, Mason, J, Naveda, F, Parrish, A & Sobel, A 2003, ‘Certification for software professionals: the IEEE computer society’s CSDP program’, paper presented to 16th Conference on Software Engineering Education and Training. (CSEE&T 2003). 20-22 March.

Sekaran, U 1992, *Research Methods for Business: A Skill Building Approach*, John Wiley & Sons, New York.

Selvanathan, EA, Selvanathan, S, Keller, G & Warrack, B 2000, *Australian Business Statistics*, 2nd edn, Nelson Thomson Learning, South Melbourne.

Sharman, R 1992, 'Quality system certification - for supplying to Government', *The Source*, no. June, p. 5.

Shaw, M 2003, 'Writing good software engineering research papers: minitutorial', in *Proceedings of the 25th International Conference on Software Engineering*, IEEE Computer Society Press, pp. 726-36.

Shewhart, WA 1931, *Economic Control of Quality of Manufactured Product*, Van Nostrand, New York.

Shull, F, Carver, J & Travassos, GH 2001, 'An empirical methodology for introducing software processes', in *Proceedings of the 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ACM Press, pp. 288-96.

Siegel, S 1956, *Nonparametric Statistics: for the Behavioral Sciences*, McGraw-Hill Book Company, Inc., Tokyo.

Sims, D 1994, 'Motorola India self-assesses at Level 5', *IEEE Software*, vol. 11, no. 2, p. 92.

Singer, J & Vinson, NG 2002, 'Ethical issues in empirical studies of software engineering', *IEEE Transactions on Software Engineering*, vol. 28, no. 12, pp. 1171-80.

Singer, J, Storey, M-A & Sim, SE 2000, 'Beg, borrow, or steal (workshop session): using multidisciplinary approaches in empirical software engineering research', in *Proceedings of the 22nd International Conference on Software Engineering*, ACM Press, pp. 799-800.

Skymark 2003, *Walter Shewhart - The Grandfather of Total Quality Management*, viewed 3 Mar 2003, <http://www.skymark.com/resources/leaders/shewart.asp>.

Slaughter, SA, Harter, DE & Krishnan, MS 1998, 'Evaluating the cost of software quality', *Communications of the ACM*, vol. 41, no. 8, pp. 67-73.

SME E-Commerce Roundtable - Thinktank 2001, *Definition of Small Business: A Small Business Coalition Initiative*, <www.setel.com.au/smeforum2002/tp/BP01.htm>.

Smillie, E 1993, 'Standards for Australia', *Informatics*, pp. 35-8.

Smith, DJ 1995, *Achieving Quality Software*, 3 edn, Chapman & Hall, London.

Smith, WL, Fletcher, RI, Gray, EM & Hunter, RB 1994, 'Software process improvement: the route to software quality?' in M Ross, CA Brebbia, G Staples & J Stapleton (eds), *Software Quality Management II Vol 1: Managing Quality Systems*, Computational Mechanics Publications, Southampton Boston, pp. 193-211.

Software Engineering Institute 2000, *Process Maturity Profile of the Software Community 2000 Mid-Year Update*, Carnegie Mellon University, Pittsburgh.

Software Quality Institute *The History of the SPICE Trials*, GU, viewed 1 October 2004, <http://www.sqi.gu.edu.au/spice/trials/p1summ.html>.

Solon, R & Statz, J 2002, 'Benchmarking the ROI for software process improvement', *The DoD SoftwareTech News*, vol. 5, no. 4, pp. 6-11.

Spector, PE 1992, *Summated Rating Scale Construction: an introduction*, Sage Publications, Newbury Park, CA.

SPICE Project 1995, *Software Process Assessment Parts 1-9, June*.

---- 1998, *SPICE Phase 2 Trials Interim Report*.

SPICE Project Team 1998, *SPICE Phase 1 Trials Report*, viewed 27 Oct 2004, <http://www.cis.strath.ac.uk/research/papers/SPICE/p1rp100pub.pdf>.

SPIRE 1998, *The Spire Handbook: Better, Faster, Cheaper Software Development in Small Organisations*, Centre for Software Engineering, DCU Campus, Dublin.

Sproull, NL 1988, *Handbook of Research Methods: A Guide for Practitioners and Students in the Social Sciences*, Scarecrow Press, Inc. Metuchen, N.J.

SQI 1999a, *Method for the Rapid Assessment of Process Capability*, Software Quality Institute Griffith University, Brisbane.

---- 1999b, *Process Model for the RAPID Assessment of Process Capability*, Software Quality Institute Griffith University, Brisbane.

Stalhane, T, Borgersen, P & Arnesen, K 1997, 'In search of the customer's quality view', *Journal of Systems Software*, vol. 38, pp. 85-93.

Stelzer, D & Mellis, W 1998, 'Success factors of organizational change in software process Improvement', *Software Process: Improvement and Practice*, vol. 4, pp. 227-50.

Storey, DJ 1994, *Understanding the Small Business Sector*, Routledge, London.

Straub, DW 1989, 'Validating instruments in MIS research', *MIS Quarterly*, vol. 13, no. 2, pp. 147-68.

Swanson, EB 1994, 'Information systems innovation among organizations', *Management Science*, vol. 40, no. 9, pp. 1069-88.

Sweeney Research 2003a, *What's Bugging Australia's Software Industry: 2003 Report Summary*, SEA National, viewed 6 Jan 2004, <http://www.seanational.com.au/articles/files/What_s_Bugging_the_Australian_Software_Industry_Report.pdf>.

---- 2003b, *What's Bugging Australia's Software Industry*, SEA National.

Tan, M & Yap, CY 1995, 'Impact of organisational maturity on software quality', in M Lee, B Barta & P Juliff (eds), *Software Quality and Productivity: Theory Practice, Education and Training*, Chapman and Hall for IFIP, London, pp. 231-4.

Taylor, FW 1911, *The principles of scientific management*, Harper & Brothers, New York.

Thomas, M & McGarry, F 1994, 'Top-down vs. bottom-up process improvement', *IEEE Software*, vol. 11, no. 4, pp. 12-3.

Thomson, H 1995, 'Aspects of Software Process Assessment and Improvement', Ph D thesis, University of East Anglia.

Thomson, H & Mayhew, P 1997, 'Approaches to software process improvement', *Software Process: Improvement and Practice*, vol. 3, pp. 3-17.

Thong, JYL, Yap, C-S & Raman, KS 1996, 'Top management support, external expertise and information systems implementation in small business', *ISR*, vol. 7, no. 2, pp. 248-67.

Thorogood, A & Yetton, P 2004, 'Reducing the technical complexity and business risk of major systems projects', paper presented to 37th Hawaii International Conference on System Sciences.

Thorwart, K 1998, 'The AMETIST process improvement experiment: towards efficient team development in small companies.' *Software Process: Improvement and Practice*, vol. 4, pp. 11-8.

Tichy, WF, Lukowicz, P, Prechelt, L & Heinz, EA 1995, 'Experimental evaluation in computer science: a quantitative study', *Journal of Systems & Software*, vol. 28, no. 1, pp. 9-18.

Tornatzky, LG & Klein, KJ 1982, 'Innovation characteristics and innovation adoption-implementation: a meta-analysis of findings', *IEEE Transactions on Engineering Management*, vol. EM-29, no. 1, pp. 28-45.

Tuffley, A, Grove, B & McNair, G 2002, 'SPICE for small organisations', paper presented to 3rd International SPICE Conference, Noordwijk, The Netherlands.

Tully, C, Kuvaja, P & Messnarz, R 1999, 'Software process analysis and improvement: a catalogue and comparison of models', in R Messnarz & C

Tully (eds), *Better Software Practice for Business Benefit: Principles and Experience*, IEEE Computer Society, pp. 51-106.

U.S. General Accounting Office 1991, *Management Practices: US Companies Improve Performance through Quality Efforts*, US General Accounting Office, Gaithersburg, MD.

Vakaslahti, P 1997, 'Process improvement frameworks: a small case study with People Capability Maturity Model', *Software Process: Improvement and Practice*, vol. 3, pp. 225-34.

Varkoi, T 2004, 'SataSPIN- a network for software companies to improve processes', paper presented to Software and Systems Standards for the Real World Seminar, Brisbane, 7 May 2004.

Vogel, DR & Wetherbe, JC 1984, 'MIS Research: a profile of leading journals and universities', *Data Base*, vol. Fall, pp. 3-14.

Voss, C, Blackmon, KL, Cagliano, R, Hanson, P & Wilson, F 1998, 'Made in Europe: small companies', *Business Strategy Review*, vol. 9, no. 4, pp. 1-19.

Wang, Y, Court, I, Ross, M, Staples, G, King, G & Dorling, A 1997, 'A survey of base process activities towards software management process excellence', paper presented to Pacific Northwest Software Quality Conference, Oregon.

Wang, Y, King, G, Dorling, A, Patel, D, Court, I, Staples, G & Ross, M 1998, 'A worldwide survey of base process activities towards software engineering process excellence', in *Proceedings of International Conference on Software Engineering*, Kyoto, Japan, pp. 439-42.

Weber, KC & do Nascimento, CJ 2002, 'Brazilian software quality in 2002', paper presented to International Conference on Software Engineering, Orlando, Florida, May 19-25.

Webster, J & Watson, RT 2002, 'Analyzing the past to prepare for the future: writing a literature review', *MIS Quarterly*, vol. 26, no. 2, pp. xiii-xxiii.

Weinberg, GM 1995, 'Just say no! Improving the requirements process', *American Programmer*, vol. 8, no. 10, pp. 19-23.

Westaway, T 1995, 'How we achieved level 3', *Crosstalk: The Journal of Defense Software Engineering*, May.

Whitehorse Strategic Group Ltd 2002, *Software Industry Research Report based on the Information Technology and Telecommunications Industry Survey December 2001*, Melbourne.

Wiegers, K 1998, 'Software process improvement: eight traps to avoid', *Crosstalk: The Journal of Defense Software Engineering*, Sept, pp. 9-12.

Wilkie, FG, McFall, D & McCaffery, F 2004, 'The centre for software process technologies: a model for process improvement in geographical regions with small software industries', paper presented to Software Engineering Process Group, Orlando, Florida, USA.

Wilson, DN, Petocz, P & Roiter, K 1995, 'A study of software quality assurance in practice', in Ross, Brebbia, Staples & Stapleton (eds), *Software Quality Management III Vol1*, pp. 159-69.

Wilson, DN, Hall, T & Baddoo, N 2001, 'A framework for evaluation and prediction of software process improvement success', *The Journal of Systems and Software*, vol. 59, pp. 135-42.

Wohlwend, H & Rosenbaum, S 1993, 'Software improvements in an international company', in *Proceedings of 15th International Conference on Software Engineering*, pp. 212-20.

---- 1994, 'Schlumberger's software improvement program', *IEEE Transactions on Software Engineering*, vol. 20, no. 11, pp. 833-9.

Wood, M, Daly, J, Miller, J & Roper, M 1999, 'Multi-method research: an investigation of object-oriented technology', *Journal of Systems and Software*, vol. 48, no. 1, pp. 13-26.

Woodhead, B 2003, 'Local software low on quality', *Financial Review*, 5 Nov, p. 48.

Wruck, KH & Jensen, MC 1994, 'Science, specific knowledge, and total quality management', *Journal of Accounting & Economics*, vol. 18, pp. 247-87.

Xia, F 1998, 'What's wrong with software engineering research methodology', *ACM SIGSOFT Software Engineering Notes*, vol. 23, no. 1, pp. 62-5.

Xydias-Lobo, M & Jones, JT 2003, *Quality Initiatives and Business Growth in Australian Manufacturing SMEs: an Exploratory Investigation*, Research Paper Series: 03-3, Flinders University.

Yamamura, G & Wigle, GB 1997, 'SEI CMM Level 5: for the right reasons', *Crosstalk: The Journal of Defense Software Engineering*, August.

Yardley, D 2002, *Successful IT Project Delivery: Learning the Lessons of Project Failure*, Addison Wesley, Harlow, UK.

Yin, RK 1994, *Case Study Research: Design and Methods*, Sage Publications, Thousand Oaks, Calif.

Zelkowitz, M & Wallace, D 1997, 'Experimental validation in software technology', *Information and Software Technology*, vol. 39, no. 11, pp. 735-44.

Zhang, Z 2000, 'Developing a model of quality management methods and evaluating their effects on business performance', *Total Quality Management*, vol. 11, no. 1, pp. 129-37.

Zikmund, WG 1994, *Business Research Methods*, Dryden Press, Fort Worth.

Zmud, RW 1982, 'Diffusion of modern software practices: influence of centralization and formalization', *Management Science*, vol. 28, no. 12, pp. 1421-31.

---- 1984, 'An examination of 'Push-Pull' theory applied to process innovation in knowledge work', *Management Science*, vol. 30, pp. 727-38.

LIST OF APPENDICES

Appendix A. List of abbreviations

Table A.1 List of Abbreviations

Abbreviation	Meaning
ABS	Australian Bureau of Statistics
ACS	Australian Computer Society
ANZSIC	Australian and New Zealand Standard Industrial Classification
BSA	Business Software Association
CBA	CMM-based appraisal
CM	Configuration management
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integrated
COTS	Commercial Off-the-Shelf
CSDP	Certified Software Development Professional
EC	European Commission
ESE	Empirical Software Engineering
ESI	European Software Institute
ESSI	European Systems and Software Initiative
FDIS	Final Draft International Standard
FTE	Full Time Equivalent
GST	Australian Government Goods and Service Taxation legislation
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IIB	Information Industries Board
IPI	Internal Process Improvement
ISBSG	International Software Benchmarking Standards Group
IS	Information Systems
ISO	International Organisation for Standardization
MIS	Management Information Systems
M-W U	Mann-Whitney U test
NIIP	National Industry Improvement Program
PA	Process attribute
PDTR	Proposed Draft Technical Report
PE	Process establishment
PIE	Process Improvement Experiment
PIP	Process Improvement Program
PM	Project management
PR	Problem resolution
PSM	Practical Software Measurement
QA	Quality assurance
Qld	Queensland
RAPID	Rapid Assessments for Process Improvement for software Development
RE	Requirements elicitation
RM	Risk management
SADT	Structured Analysis and Design Techniques
SBPQ	Software Best Practices Questionnaire
SCAMPI	Standard CMMI SM Appraisal Method for Process Improvement
SCE	Software Capability Evaluation
SD	Software development
SE	Software Engineering

SEA	Software Engineering Australia
SEAQ	Software Engineering Australia – Queensland Branch
SEI	Software Engineering Institute
SME	Small Medium Enterprise
SPA	Software Process Assessment
SPC	Statistical Process Control
SPI	Software Process Improvement
SPICE	Software Process Improvement and Capability dEtermination
SQA	Software Quality Assurance
SQI	Software Quality Institute
TR	Technical Report
VASIE	Value Added Software Information for Europe

Appendix B. Glossary of statistical tests and terminology

Data scale types. The statistical operations allowable on a given set of scores are dependent on the level of measurement achieved. The four levels of measurement are as follows: nominal (qualitative; categorical observations), ordinal (ranked categorical), interval (quantitative), and ratio (has true zero point) (Selvanathan et al. 2000, pp. 14-6; Siegel 1956).

***p*-value** is the 'probability that the observed data or data more extreme, given that the null hypothesis is true, and the sampling was done randomly' (Miller 2004, p. 185) and is referred in the statistical tables as *p*. The *p*-value is compared to the significance criterion α. In this study, α is set at 0.05. The null hypothesis is rejected if the p-value is less than α.

Independent groups t-test is used to confirm if interval or ratio data represents two populations. The t-test is a parametric test, and assumes the data is drawn from two independent groups and that the two populations have equal variances. Levene's test is used to compare the variances of the two populations.

Normality is a prerequisite for some statistical tests. To test if the data arise from a normal distribution, the Shapiro-Wilks W-statistic is used if the sample size is <=50, and the Kolmogorov D statistic if the sample size is > 50 (Bonate 2000, p. 58; Coakes & Steed 1996).

An **Outlier** is an observation that is unusually small or unusually large (Selvanathan et al. 2000, p. 662). Observations with values between 1.5 and 3 box lengths from the upper or lower edge of the box are referred to as outliers. The box length is the interquartile range. **Extremes** are observations with values more than 3 box lengths from the upper or lower edge of the box. Source: SPSS Help (SPSS Inc. 2004)

The **One-Way ANOVA** procedure produces a one-way analysis of variance for a quantitative dependent variable by a single factor (independent) variable. Analysis of variance is used to test the hypothesis that several means are equal. This technique is an extension of the two-sample t test. In addition to determining that differences exist among the means, this test can identify which means differ. There are two types of tests for comparing means: a priori contrasts and post hoc tests. Contrasts are tests set up before running the experiment, and post hoc tests are run after the experiment has been conducted. Source: SPSS Help (SPSS Inc. 2004)

ANOVA was used to compare the adoption level of organisations within each industry sector. As population normality is a prerequisite assumption for analysis of variance, the calculation was performed separately for the non-COTS developers and for the COTS developers. As the second assumption for the ANOVA test is homogeneity of variance, Levene's test was performed in each case.

Friedman test is the nonparametric equivalent of a one-sample repeated measures design or a two-way analysis of variance with one observation per cell. Friedman tests the null hypothesis that k related variables come from the same population. For each case, the k variables are ranked from 1 to k. The test statistic is based on these ranks. Kendall's W is a normalization of the Friedman statistic. Kendall's W is interpretable as the coefficient of concordance, which is a measure of agreement among raters. Each case is a judge or rater and each variable is an item or person being judged. For each variable, the sum of ranks is computed. Kendall's W ranges between 0 (no agreement) and 1 (complete agreement). Source: SPSS Help (SPSS Inc. 2004).

Wilcoxon signed ranks test is a paired-sample test. It is the non-parametric equivalent to the paired samples t-test. The only assumption is that the sample distributions are symmetric about the median and the number of tied ranks is small. The test involves calculating the difference scores from high to low, affixing the sign of each difference to the corresponding rank. In the case of tied ranks, the mean of the rank which would have been assigned to those observations had they not been tied, is used (Bonate 2000, p. 58; Selvanathan et al. 2000, p. 443). The Wilcoxon pretest/posttest was used for pair-wise comparisons to determine if a statistical difference exists between the capability levels at assessment and follow-up meeting.

The **Mann-Whitney U test** is used to determine whether two independent samples (groups) come from the same population. It is the most popular of the two-independent-samples tests. It is equivalent to the Wilcoxon rank sum test and the Kruskal-Wallis test for two groups. Mann-Whitney tests that two sampled populations are equivalent in location. The observations from both groups are combined and ranked, with the average rank assigned in the case of ties. The number of ties should be small relative to the total number of observations. If the populations are identical in location, the ranks should be randomly mixed between the two samples. The number of times a score from group 1 precedes a score from group 2 and the number of times a score from group 2 precedes a score from group 1 are calculated. The Mann-Whitney U statistic is the smaller of these two numbers. The Wilcoxon rank sum W statistic, also displayed, is the rank sum of the smaller sample. If both samples have the same number of observations, W is the rank sum of the group named. From SPSS Help (SPSS Inc. 2004). In some of the tables in this study, referred to as M-W U.

Pearson coefficient of correlation (also known as Pearson product-moment correlation) is a parametric test to measure the strength of association between two variables (Selvanathan et al. 2000) . The prerequisites are that the data must be collected from related pairs; the scale of measurement should be interval or ratio; scores within each variable should be normally distributed; the relationship between the 2 variables must be linear; the variability in scores for the variables is roughly the same (Coakes & Steed 1996, p. 49-50).

Spearman rank correlation coefficient (rho) is used to measure and test if a relationship exists between two variables (Selvanathan et al. 2000, p. 653). It is the non-parametric alternative to the parametric bivariate correlation (Pearson's r) (Coakes & Steed 1996, p. 166). In this study, the Spearman rank correlation coefficient (rho) is labelled r_s.

Kruskall-Wallis test is a non-parametric equivalent of independent-samples single-factor analysis of variance. The K-W test can be applied when the problem objective is to compare two or more populations; the data are either ranked or quantitative but non-normal; the samples are independent (Selvanathan et al. 2000, p. 564). It is used in this study to compare process capability according to target business sector.

Discriminate function analysis is used to determine which continuous variables discriminate between two or more naturally occurring groups. It answers the question: can a combination of variables be used to predict group membership? In discriminant analysis, the independent variables are the predictors and the dependent variables are the groups. Usually, several variables are included in a study to see which ones contribute to the discrimination between groups. Wilks' lambda is used in an ANOVA (F) test of mean differences in discriminant analysis, such that the smaller the lambda for an independent variable, the more that variable contributes to the discriminant function. Lambda varies from 0 to 1, with 0 meaning group means differ (thus the more the variable differentiates the groups), and 1 meaning all group means are the same. The F test of Wilks' lambda shows which variables' contributions are significant. Variables should be continuous and normally distributed (Poulsen & French 2002).

Fisher's exact probability test is useful for analysing either nominal or ordinal discrete data when the two independent samples are small in size. The test determines whether the two groups differ in the proportion with which they fall into the two classifications. Fisher's exact test if used in this research to compare responses of high and low adopters.

Bonferroni correction method. The probability of committing a Type I error (falsely reject H_0) increases with the number of tests. To overcome this risk, the Bonferroni correction method provides a simple corrective procedure by dividing the desired α level by the number of tests involved (Miller 2004, p. 184).

Appendix C. Timeframe of study

Table C.1 List of research activities with start and end dates

Activity	From date	To date
Literature Review	Jul 1998	Nov 2004
Modify Questionnaire	Sep 1998	Jan 1999
Pre-test, pilot test	Dec 1998	Jan 1999
Mail out questionnaire	Jan 1999	Jan 1999
Preliminary analysis of survey responses	Feb 1999	Mar 1999
Prepare preliminary report for SEA'99	Apr 1999	
Further survey analysis	Jan 2000	Mar 2000
Prepare ASWEC paper	Mar 2000	April 2000
Participate in PIP assessments	Oct 1999	Dec 1999
Prepare PIP assessment reports	Jan 2000	Mar 2000
Plan and conduct follow-up meetings	June 2000	Aug 2000
Prepare final reports	Aug 2000	Sep 2000
Preliminary analysis of four PIP firms	Sep 2000	Dec 2000
Further analysis of survey data	Jan 2001	Nov 2004
Further analysis of PIP reports	Jan 2001	Nov 2004
Comparison survey and PIP results	Jan 2001	Nov 2004
Write and review thesis	Jan 2003	Dec 2004

Appendix D. Mappings related to survey

Table D.1 Mapping of Best Practice Survey Questions to ESI Questionnaire

ESSI	ESI question	ESI section		New question for this research	New section	
	Organisation Type	General Information		Deleted		
	Country	General Information		Deleted		
	Number of Employees	General Information		A4	Your Organisation	A
	Number of Employees involved in Software Engineering	General Information		A5	Your Organisation	A
	Organisation primary involvement in software industry	General Information		A3	Your Organisation	A
	Industrial Sector	General Information		A2	Your Organisation	A
	Community of Interest	General Information		Deleted		
				A1 Contact information	Your Organisation	A
				A6	Your Organisation	A
				A7	Your Organisation	A
1.1	1.1 Does each software project have a nominated software project manager?	Organisational Issues	1	5.1	Project Management & Training	5
1.2	1.2 Does the software project manager report to a business project manager responsible for the overall benefit of the project to the business?	Organisational Issues	1	5.2	Project Management & Training	5
1.3	1.3 Does a Software Quality Assurance (SQA) function exist within an independent reporting line from software development project management?	Organisational Issues	1	5.3 Does a Software Quality Assurance (SQA) function exist? If YES, does the SQA function have an independent reporting	Project Management & Training	5

ESSI	ESI question	ESI section		New question for this research	New section	
				line from the software development project management?		
1.4	1.4 Is a change control function established for each software project?	Organisational Issues	1	3.1	Configuration Management	5
1.5	1.5 Is there a required training programme for all newly-appointed software managers which is designed to familiarise them with in-house software project management procedures?	Organisational Issues	1	5.9	Project Management & Training	5
1.6	1.6 Is there a procedure for maintaining awareness of the state-of-the-art in CASE or software engineering technology?	Organisational Issues	1	5.10	Project Management & Training	5
1.7	1.7 Is there a procedure for ensuring that appropriate levels of user/customer/marketing input is made throughout the project?	Organisational Issues	1	1.1	Requirements and Design	1
1.8	(1.8) Where other non-software resources are critical to the success of the project is there a procedure for ensuring their availability according to plan?	Organisational Issues	1	4.9	Estimates and Schedules	4
2.01	2.1 Do management formally assess the benefits, viability, and risk of each software project prior to making contractual (or internal) commitments?	Standards and Procedures	2	5.4	Project management & Training	5
2.02	2.2 Do management formally conduct periodic reviews of the status of each software project?	Standards and Procedures	2	5.6	Project Management & Training	5
2.03	(2.3) Are there procedures to ensure that external software subcontracting organisations, if any, follow a disciplined software development process?	Standards and Procedures	2	(5.5)	Project Management & Training	5
2.04	2.4 For each project, are independent audits (such as inspections or walkthroughs) conducted for each major stage in the software development process?	Standards and Procedures	2	5.7	Project Management & Training	5
2.05	2.5 Are common coding standards applied to each software	Standards and	2	2.4	Code and Test	2

ESSI	ESI question	ESI section		New question for this research	New section	
	project?	Procedures				
2.06	2.6 Is there a documented procedure for estimating software size (such as 'Lines of Source Code') and thus for using productivity measures?	Standards and Procedures	2	4.1	Estimates and Schedules	4
2.07	2.7 Is a formal procedure used to produce software development effort, schedule, and cost estimates?	Standards and Procedures	2	4.2	Estimates and Schedules	4
2.08	2.8 Is a formal procedure (such as a review or handover with sign-off) used whenever a deliverable (such as a user statement of requirements or system requirements) is passed from one discrete group to another (e.g. user to analyst to designer) to ensure it is properly understood?	Standards and Procedures	2	5.8	Project Management & Training	5
2.09	(2.9) Is there a procedure to ensure that the systems projects selected for development qualitatively or quantitatively support/alleviate the organisation's business objective/problems?	Standards and Procedures	2	(1.2)	Requirements and Design	1
2.10	(2.10) Are there procedures to ensure that the functionality, strengths, and weaknesses of the 'system' which the software is replacing are formally reviewed?	Standards and Procedures	2	1.4	Requirements and Design	1
2.11	2.11 Does test planning commence prior to programming beginning based on the user requirements and high-level design documents?	Standards and Procedures	2	2.2	Code and Test	2
2.12	2.12 Is independent testing conducted by users (or appropriate representatives) under the guidance of Software Quality Assurance before any system or enhancement goes live?	Standards and Procedures	2	2.5 Is independent testing conducted by users (or appropriate representatives) before any system or enhancement goes live? If YES, is it under the guidance of Software Quality Assurance?	Code and Test	2
2.13	2.13 Is there a procedure to check that the system configuration (i.e. the programs and any data) passing user acceptance testing is the same as that which is implemented	Standards and Procedures	2	2.6 Is there a procedure to check that the system configuration (i.e. the programs and any data)	Code and Test	2

ESSI	ESI question	ESI section		New question for this research	New section	
	for live operation and that no changes are made directly to a 'live' version of any system (other than through modification to its development version)?			passing user acceptance testing is the same as that which is implemented for live operation? Is there a procedure to check that no changes are made directly to a 'live' version of any system (other than through modification to its development version)?		
3.1	3.1 Are records of actual project resourcing and timescales versus estimates maintained (at individual resource/resource-type level) and regularly analysed/fed-back into the estimating and scheduling procedures?	Metrics	3	4.3	Estimates and Schedules	4
3.2	(3.2) Are records of software size maintained for each software configuration item, over time, and fed-back into the estimating process?	Metrics	3	(4.4)	Estimates and Schedules	4
3.3	3.3 Are statistics on the sources of errors in software code gathered and analysed for their cause, detection and avoidance measures?	Metrics	3	2.12	Code and Test	2
3.4	3.4 Are statistics on test efficiency (% of errors actually detected by an activity against the maximum theoretically possible) gathered and analysed for all testing stages in the development process?	Metrics	3	2.9 Are statistics on test efficiency (**eg.** % of errors actually detected by an activity against the maximum theoretically possible) gathered and analysed for all testing stages in the development process?	Code and Test	2
3.5	3.5 Is project tracking (e.g. earned value) used throughout the software development process (actual versus planned deliverables analyses, designed, unit tested, system tested, acceptance tested over time) to monitor project progress?	Metrics	3	4.5	Estimates and Schedules	4
3.6	(3.6) Are estimates made and compared with actuals for target computer performance (e.g. memory utilisation,	Metrics	3	(2.8)	Code & Test	2

ESSI	ESI question	ESI section		New question for this research	New section	
	processor throughput and file/channel I/O and disk usage)?					
3.7	3.7 Are post-implementation software problem reports logged and their resolution effectively tracked and analysed?	Metrics	3	2.10	Code and Test	2
3.8	3.8 Do records exist from which (and requiring nothing extra) all current versions and variants of software systems and their components can be quickly and accurately reconstructed in the development environment?	Metrics	3	2.11	Code and Test	2
4.1	4.1 Are estimates, schedules and subsequent changes produced only by the project managers who directly control the project resources and are fully aware of their abilities and availabilities?	Control of the Development Process	4	4.6	Estimates and Schedules	4
4.2	4.2 Does the overall business project manager gain agreement and sign-off from all parties who have produced detailed estimates and schedules before publishing or revising a consolidated project plan?	Control of the Development Process	4	4.8	Estimates and Schedules	4
4.3	4.3 Is there a procedure for controlling changes to the software requirements, designs and accompanying documentation?	Control of the Development Process	4	3.2	Configuration Management	3
4.4	4.4 Is there a procedure for controlling changes to the code and specifications?	Control of the Development Process	4	3.3	Configuration Management	3
4.5	4.5 Is there a procedure for assuring that regression testing (i.e. the forced re-run of all previous tests prior to any new tests) is routinely performed during and after initial implementation?	Control of the Development Process	4	2.7	Code and Test	2
4.6	4.6 Do procedures exist to ensure that every required function is tested/verified?	Control of the Development Process	4	2.13	Code and Test	2
5.1	5.1 Are software tools used to assist in forwards and/or backwards tracing of software requirements to software	Tools and Technology	5	3.4	Configuration Management	3

ESSI	ESI question	ESI section		New question for this research	New section	
	designs through to code?					
5.2	5.2 Are design notations such as Structured Analysis and Design Technique used in program design?	Tools and Technology	5	1.3	Requirements and Design	1
5.3	5.3 Are automated testing tools used (for example for capturing and replaying tests, or for ensuring logic paths coverage)?	Tools and Technology	5	Included in new Q 2.3	Code and Test	2
5.4	5.4 Are software tools used for tracking and reporting the status of the software/subroutines in the software development library?	Tools and Technology	5	3.5 Are software tools used for tracking and reporting the status (eg. reviewed, tested, released) of the software/subroutines in the software development library?	Configuration Management	3
5.5	5.5 Are prototyping methods used in ensuring the requirements elements of the software?	Tools and Technology	5	1.5	Requirements and Design	1
5.6	(5.6) Is a data dictionary available for controlling and storing details of all data files and their fields?	Tools and Technology	5	(1.6)	Requirements and Design	1
5.7	5.7 Are software tools used for project planning, estimating, scheduling, and critical path analysis?	Tools and Technology	5	4.7	Estimates & Schedules	4
				2.1 Programming languages used	Code & Test	2
				2.3 Development tools used	Code & Test	2
				6.1 Which aspects of your software development activities have the most for improvement?	How can we help you?	6
				6.2 How can SEA help you achieve those improvements?	How can we help you?	6

Note: ESI survey questions sourced from ESI (1995)

Table D.2 Mapping of ANZSIC division codes to industry sectors used by ESI

ANZSIC Code	ANZSIC Division	ESI Industry sectors
A	Agriculture, forestry & fishing	Agriculture & forestry
		Fishing
B	Mining	Mining & quarrying
C	Manufacturing	17 manufacturing industries
D	Electricity, gas & water supply	Energy production & distribution; gas & water supply
E	Construction	Construction & building
F	Wholesale trade	Wholesale & retail trade; repair of goods
G	Retail trade	
H	Accommodation, cafes & restaurants	Lodging & restaurants
I	Transport & storage	Transportation services
J	Communication services	Post & telecommunications
		Publishing, printing and reproduction of recorded media
K	Finance & insurance	Finance & insurance
L	Property & business services	Business, legal and management consultancy; holdings
		Real estate activities
		Renting & leasing
M	Government administration & defence	
N	Education	Education
O	Health & community services	Community service activities
		Health & social work
P	Cultural & recreational services	Recreational, cultural and sporting activities
Q	Personal & other services	
		6 IT activities
		Technical testing & analysis
		Recycling
		Electrical engineering and related technical consultancy
		Mechanical engineering and related technical consultancy

Table D.3 Mapping of Best Practice Survey to ISO/IEC TR 15504 and RAPID model

Section 1—Requirements and Design	ISO/IEC 15504	RAPID
1.1 Is there a procedure for ensuring that appropriate levels of user/customer/marketing input are made throughout the project?	CUS.3 Requirements elicitation	RE
1.2 Is there a procedure to ensure that the systems projects selected for development qualitatively or quantitatively support/alleviate the organisation's business objective/problems?	CUS.1.1 Acquisition preparation	No
1.3 Are design notations such as Structured Analysis and Design Technique used in program design?	ENG.1.3 Software design	SD
1.4 Are there procedures to ensure that the functionality, strengths, and weaknesses of the 'system' which the software is replacing are formally reviewed?	No	No
1.5 Are prototyping methods used in ensuring the requirements elements of the software?	ENG.1.1 Systems requirements analysis and design	SD
1.6 Is a data dictionary available for controlling and storing details of all data files and their fields?	ENG.1.3 Software design	SD
Section 2—Code and Test	ISO/IEC 15504	RAPID
2.2 Does test planning commence prior to programming beginning based on the user requirements and high-level design documents?	ENG.1.6 Software testing	SD
2.4 Are common coding standards applied to each software project?	ENG.1.4 Software construction	SD
2.5a Is independent testing conducted by users (or appropriate representatives) before any system or enhancement goes live?	ENG.1.5 Software integration	SD
2.5b If YES, is it under the guidance of Software Quality Assurance?	SUP.3 Quality assurance	QA
2.6a Is there a procedure to check that the system configuration (i.e. the programs and any data) passing user acceptance testing is the same as that which is implemented for live operation?	SUP.2 Configuration management	CM
2.6b Is there a procedure to check that no changes are made directly to a 'live' version of any system (other than through modification to its development version)?	SUP.2 Configuration management	CM
2.7 Is there a procedure for assuring that regression testing (i.e. the forced re-run of all previous tests prior to any new tests) is routinely performed during and after initial implementation?	ENG.1.6 Software testing	SD
2.8 Are estimates made and compared with actuals for target computer performance (e.g. memory utilisation, processor throughput and file/channel I/O and disk usage)?	MAN.3 Quality management	No
2.9 Are statistics on test efficiency (eg. % of errors actually detected by an activity against the maximum theoretically possible) gathered and analysed for all testing stages in the development process?	ENG.1.6 Software testing	SD

Section 2—Code and Test (continued)	ISO/IEC 15504	RAPID
2.10 Are post-implementation software problem reports logged and their resolution effectively tracked and analysed?	SUP.8 Problem resolution	PR
2.11 Do records exist from which (and requiring nothing extra) all current versions and variants of software systems and their components can be quickly and accurately reconstructed in the development environment?	SUP.2 Configuration management	CM
2.12 Are statistics on the sources of errors in software code gathered and analysed for their cause, detection and avoidance measures?	SUP.4 Verification	No
2.13 Do procedures exist to ensure that every required function is tested/verified?	SUP.4 Verification	No
Section 3—Configuration Management	ISO/IEC 15504	RAPID
3.1 Is a change control function established for each software project?	SUP.2 Configuration management	CM
3.2 Is there a procedure for controlling changes to the software requirements, designs and accompanying documentation?	SUP.2 Configuration management	CM
3.3 Is there a procedure for controlling changes to the code and specifications?	SUP.2 Configuration management	CM
3.4 Are software tools used to assist in forwards and/or backwards tracing of software requirements to software designs through to code?	ENG.1 Development	SD
3.5 Are software tools used for tracking and reporting the status (eg. reviewed, tested, released) of the software/subroutines in the software development library?	SUP.2 but not software tools	CM
Section 4—Estimates and Schedules	ISO/IEC 15504	RAPID
4.1 Is there a documented procedure for estimating software size (such as ‘Lines of Source Code’) and thus for using productivity measures?	MAN.2 Project management	PM
4.2 Is a formal procedure used to produce software development effort, schedule, and cost estimates?	MAN.2 Project management	PM
4.3 Are records of actual project resourcing and timescales versus estimates maintained (at individual resource/resource-type level) and regularly analysed/fed-back into the estimating and scheduling procedures?	MAN.2 Project management – first part	PM
4.4 Are records of software size maintained for each software configuration item, over time, and fed-back into the estimating process?	SUP.2 Configuration management	CM
4.5 Is project tracking (e.g. earned value) used throughout the software development process (actual versus planned deliverables analyses, designed, unit tested, system tested, acceptance tested over time) to monitor project progress?	MAN.2 Project management	PM
4.6 Are estimates, schedules and subsequent changes produced only by the project managers who directly control the project resources and are fully aware of their abilities and availabilities?	MAN.2 Project management	PM

Section 4—Estimates and Schedules (continued)	ISO/IEC 15504	RAPID
4.7 Are software tools used for project planning, estimating, scheduling, and critical path analysis?	MAN.2 Project management	PM
4.8 Does the overall business project manager gain agreement and sign-off from all parties who have produced detailed estimates and schedules before publishing or revising a consolidated project plan?	MAN.2 Project management	PM
4.9 Where other non-software resources are critical to the success of the project is there a procedure for ensuring their availability according to plan?	MAN.2 Project management	PM
Section 5—Project Management & Training	ISO/IEC 15504	RAPID
5.1 Does each software project have a nominated software project manager?	MAN.2 Project management	PM
5.2 Does the software project manager report to a business project manager responsible for the overall benefit of the project to the business?	MAN.2 Project management	PM
5.3a Does a Software Quality Assurance (SQA) function exist?	SUP.3 Quality assurance	QA
5.3b If YES, does the SQA function have an independent reporting line from software development project management?	SUP.3 Quality assurance	QA
5.4 Do management formally assess the benefits, viability, and risk of each software project prior to making contractual (or internal) commitments?	MAN.2 Project management	PM
5.5 Are there procedures to ensure that external software subcontracting organisations, if any, follow a disciplined software development process?	CUS.1.3 Supplier monitoring	No
5.6 Do management formally conduct periodic reviews of the status of each software project?	MAN.2 Project management	PM
5.7 For each project, are independent audits (such as inspections or walkthroughs) conducted for each major stage in the software development process?	SUP.3 Quality assurance	QA
5.8 Is a formal procedure (such as a review or handover with sign-off) used whenever a deliverable (such as a user statement of requirements or system requirements) is passed from one discrete group to another (e.g. *user* to *analyst* to *designer*) to ensure it is properly understood?	SUP.3 Quality assurance	QA
5.9 Is there a required training programme for all newly-appointed software managers which is designed to familiarise them with in-house software project management procedures?	ORG.3 Human resource management	No
5.10 Is there a procedure for maintaining awareness of the state-of-the-art in CASE or software engineering technology?	ORG.4 Infrastructure	No

Table D.4 ISO/IEC 15504 Processes

Process group	Basic processes	Component processes
Primary life cycle processes		
Customer	CUS.1 Acquisition	CUS 1.1 Acquisition preparation
		CUS 1.2 Supplier selection
		CUS 1.3 Supplier monitoring
		CUS 1.4 Customer acceptance
	CUS.2 Supply	
	CUS.3 Requirements elicitation	
	CUS.4 Operation	CUS 4.1 Operational use
		CUS 4.2 Customer support
Engineering	ENG.1 Development	ENG 1.1 System requirements analysis and design
		ENG 1.2 Software requirements analysis
		ENG 1.3 Software design
		ENG 1.4 Software construction
		ENG 1.5 Software integration
		ENG 1.6 Software testing
		ENG 1.7 System integration and testing
	ENG.2 System and software maintenance	
Supporting life cycle processes		
Support	SUP.1 Documentation	
	SUP.2 Configuration management	
	SUP.3 Quality assurance	
	SUP.4 Verification	
	SUP.5 Validation	
	SUP.6 Joint review	
	SUP.7 Audit	
	SUP.8 Problem resolution	
Organisational life cycle processes		
Management	MAN.1 Management	
	MAN.2 Project management	
	MAN.3 Quality management	
	MAN.4 Risk management	
Organisation	ORG.1 Organisational alignment	
	ORG.2 Improvement	ORG 2.1 Process establishment
		ORG 2.2 Process assessment
		ORG 2.3 Process improvement
	ORG.3 Human resource management	
	ORG.4 Infrastructure	
	ORG.5 Measurement	
	ORG.6 Reuse	

Note: There are 24 basic processes and 16 component processes, arranged in a 4 level hierarchy. At the top level, the three principal groupings are defined in ISO/IEC 12207 as primary, supporting and organisational life cycle processes.
Sourced from figure 1 (ISO/IEC TR 15504-2 1998, p. 5).

Appendix E. Survey Cover Letter and Questionnaire

Software Development Manager
Company
Street
Suburb
State Postcode

Dear Sir/Madam

Software Engineering Australia is undertaking an industry improvement program for the software industry. The goal of the initiative is to improve the capability and competitiveness of the Australian software development industry. In the first phase of the program, the level of use of widely-recognised best practices in the software industry will be established through the use of the accompanying survey. The survey, developed by the European Software Institute has been adapted for use in Australia to enable us to tailor the program to industry's needs. Analysis of the survey results will enable a comparison of local industry with international developers, and the identification of potential candidates for improvement projects.

I would appreciate it if you could use a small amount of your valuable time to respond to the survey. The survey is being sent to all commercial software developers and the large organisations involved in software development in Queensland. Total confidentiality is assured. The results will be summarised; individual responses will not be published. After the responses have been analysed, a summary report will be mailed to all respondents.

The survey has been designed to minimise the time demands on participants, and mostly requires indicating a response by placing a tick in the appropriate box. It should take around 10 minutes to fill out. I would appreciate your completing the survey and returning it in the reply-paid envelope by 30 January 1999.

Should you have any queries about this survey, please do not hesitate to contact me on (07) 3236 1111.

Thank you for your participation.

Phil Scanlan
Chief Executive Officer

Software Survey (adapted from ESSI)

Part A: Your Organisation

This section is related to the type and size of your organisation (that is the Queensland-based division or business unit that you belong to).

A1. Contact Information: *please complete or attach a business card.*

Your name ____________________

Organisation Name ____________________

Postal Address ____________________

____

Postcode ______ Phone ______ Fax ______

Email Address ____________________

A2. Is your organisation a software development company?

- ☐ Yes
- ☐ No

If not, which industry sector does your organisation operate in? *(Please tick one box)*

- ☐ Agriculture, Forestry & Fishing
- ☐ Mining
- ☐ Manufacturing
- ☐ Utilities - Electricity, Gas & Water
- ☐ Construction
- ☐ Retail & Wholesale
- ☐ Accommodation, Cafes & Restaurants
- ☐ Transport & Storage
- ☐ Communication Services & Media
- ☐ Finance & Insurance
- ☐ Property & Business Services
- ☐ Government Administration & Defence
- ☐ Education
- ☐ Health & Community Services
- ☐ Cultural & Recreational Services
- ☐ Personal & Other Services
- ☐ Tourism & Hospitality
- ☐ Information Technology
- ☐ Other (please state): ____________________

A3. Organisation primary involvement in software industry *(please tick appropriate box(es))*

- ☐ Software user (developed in-house)
- ☐ Software user (developed by a 3rd party)
- ☐ Software developer (producing off-the-shelf systems)
- ☐ Software developer (producing custom software systems)
- ☐ Research & Development institute or university
- ☐ Interest Group (e.g. professional society or standards body)
- ☐ Other *(please specify)*: ____________________

A4. Number of Employees __________

A5. Number of Employees involved in Software Development or Maintenance __________

A6. Would your organisation be interested in participating in the Software Engineering Australia improvement program? *(Please tick one box)*

- ☐ Yes
- ☐ No

A7. Does your organisation develop or maintain software? *(Please tick one box)*

- ☐ Yes
- ☐ No

If the answer to question A7 is 'no', then no further answers are required; please return the survey in the reply paid envelope.

Part B: Software Practices

Guidelines

1. Please attempt to answer ALL questions either YES, NO, or N.A. (Not Applicable). N.A. may be a valid response for those questions which are applicable for only certain types of developer. Questions of this type are indicated by brackets around the number of the question, e.g.(1.8), and by an "Applies to" comment set with italics.
2. "Don't know" or "don't understand the question" responses should be reflected as a NO.
3. Please answer according to normal organisational practice - not ideal practice or according to unimplemented standards.

Section 1 - Requirements and Design

1.1 Is there a procedure for ensuring that appropriate levels of user/customer/marketing input is made throughout the project?

- ☐ Yes
- ☐ No

Comments:

(1.2) Is there a procedure to ensure that the systems projects selected for development qualitatively or quantitatively support/alleviate the organisation's business objective/ problems?

- ☐ Yes
- ☐ No
- ☐ N.A.

Applies to: those organisations where all software projects do not have to be funded externally such as In-House systems and Package development. In an end-user organisation this discipline is often called Information Systems Planning.

Comments:

1.3 Are design notations such as Structured Analysis and Design Technique used in program design?

- ☐ Yes
- ☐ No

Comments:

Qld Jan

Software Survey (adapted from ESSI)

(1.4) Are there procedures to ensure that the functionality, strengths, and weaknesses of the "system" which the software is replacing are formally reviewed?

☐ Yes
☐ No
☐ N.A.

Applies to: Organisations whose software is intended to replace a previously computer-based mechanical or clerical set of tasks.

Comments:

1.5 Are prototyping methods used in ensuring the requirements elements of the software?

☐ Yes
☐ No

Comments:

(1.6) Is a data dictionary available for controlling and storing details of all data files and their fields?

☐ Yes
☐ No
☐ N.A.

Applies to: Where systems developed have significant file/database content.

Comments:

Section 2 - Code and Test

2.1 Programming language(s) used now and planned in the next 12 months *(Please tick appropriate box(es))*

NOW		NEXT 12 MONTHS
☐	C	☐
☐	C++	☐
☐	COBOL	☐
☐	Delphi	☐
☐	Easytrieve	☐
☐	Java	☐
☐	Ms-Access	☐
☐	Natural	☐
☐	Oracle	☐
☐	PL/I	☐
☐	Powerbuilder	☐
☐	SQL	☐
☐	Telon	☐
☐	Visual Basic	☐
☐	Other (please specify)	☐

2.2 Does test planning commence prior to programming beginning based on the user requirements and high-level design documents?

☐ Yes
☐ No

Comments:

2.3 Development tools used now and planned in the next 12 months *(Please tick appropriate box(es))*

NOW		NEXT 12 MONTHS
☐	Defect tracking (eg TestTrack, DDTS)	☐
☐	Source code control (eg RCS, Source Safe)	☐
☐	Automated test capture (eg Visual Test)	☐
☐	Automated code analysis (eg Code Check)	☐
☐	Code coverage (eg Pure Coverage)	☐
☐	3rd party reuseable components	☐
☐	Staff time-sheet system	☐
☐	Other (please specify)	☐

2.4 Are common coding standards applied to each software project?

☐ Yes
☐ No

Comments:

2.5 Is independent testing conducted by users (or appropriate representatives) before any system or enhancement goes live?

☐ Yes
☐ No

If YES, is it under the guidance of Software Quality Assurance?

☐ Yes
☐ No

Comments:

2.6 Is there a procedure to check that the system configuration (i.e. the programs and any data) passing user acceptance testing is the same as that which is implemented for live operation?

☐ Yes
☐ No

Is there a procedure to check that no changes are made directly to a "live" version of any system (other than through modification to its development version)?

☐ Yes
☐ No

Comments:

2.7 Is there a procedure for assuring that regression testing (i.e. the forced re-run of all previous tests prior to any new tests) is routinely performed during and after initial implementation?

☐ Yes
☐ No

Comments:

(2.8) Are estimates made and compared with actuals for target computer performance (e.g. memory utilisation, processor throughput and file/channel I/O and disk usage)?

☐ Yes
☐ No
☐ N.A.

Applies to: Where the user/customer has explicit or implicit performance expectations or computer resource constraints.

Comments:

 Qld Jan

Software Survey (adapted from ESSI)

2.9 Are statistics on test efficiency (eg. % of errors actually detected by an activity against the maximum theoretically possible) gathered and analysed for all testing stages in the development process?
☐ Yes
☐ No

Comments:

2.10 Are post-implementation software problem reports logged and their resolution effectively tracked and analysed?
☐ Yes
☐ No

Comments:

2.11 Do records exist from which (and requiring nothing extra) all current versions and variants of software systems and their components can be quickly and accurately reconstructed in the development environment?
☐ Yes
☐ No

Comments:

2.12 Are statistics on the sources of errors in software code gathered and analysed for their cause, detection and avoidance measures?
☐ Yes
☐ No

Comments:

2.13 Do procedures exist to ensure that every required function is tested/verified?
☐ Yes
☐ No

Comments:

Section 3 - Configuration Management

3.1 Is a change control function established for each software project?
☐ Yes
☐ No

Comments:

3.2 Is there a procedure for controlling changes to the software requirements, designs and accompanying documentation?
☐ Yes
☐ No

Comments:

3.3 Is there a procedure for controlling changes to the code and specifications?
☐ Yes
☐ No

Comments:

3.4 Are software tools used to assist in forwards and/or backwards tracing of software requirements to software designs through to code?
☐ Yes
☐ No

Comments:

3.5 Are software tools used for tracking and reporting the status (eg. reviewed, tested, released) of the software/subroutines in the software development library?
☐ Yes
☐ No

Comments:

Section 4 - Estimates and Schedules

4.1 Is there a documented procedure for estimating software size (such as "Lines of Source Code") and thus for using productivity measures?
☐ Yes
☐ No

Comments:

4.2 Is a formal procedure used to produce software development effort, schedule, and cost estimates?
☐ Yes
☐ No

Comments:

4.3 Are records of actual project resourcing and timescales versus estimates maintained (at individual resource/resource-type level) and regularly analysed/fed-back into the estimating and scheduling procedures?
☐ Yes
☐ No

Comments:

(4.4) Are records of software size maintained for each software configuration item, over time, and fed-back into the estimating process?
☐ Yes
☐ No

Comments:

4.5 Is project tracking (e.g. earned value) used throughout the software development process (actual versus planned deliverables analyses, designed, unit tested, system tested, acceptance tested over time) to monitor project progress?
☐ Yes
☐ No

Comments:

4.6 Are estimates, schedules and subsequent changes produced only by the project managers who directly control the project resources and are fully aware of their abilities and availabilities?
☐ Yes
☐ No

Comments:

4.7 Are software tools used for project planning, estimating, scheduling, and critical path analysis?
☐ Yes
☐ No

Comments:

4.8 Does the overall business project manager gain agreement and sign-off from all parties who have produced detailed estimates and schedules before publishing or revising a consolidated project plan?
☐ Yes
☐ No

Comments:

Qld Jan

Software Survey (adapted from ESSI)

(4.9) Where other non-software resources are critical to the success of the project is there a procedure for ensuring their availability according to plan?

☐ Yes
☐ No
☐ N.A.

Applies to: Organisations with projects dependent on a number of non-software resources such as hardware design specialists, computer operations staff, data administrators, or computer network staff.

Comments:

Section 5 - Project Management & Training

5.1 Does each software project have a nominated software project manager?

☐ Yes
☐ No

Comments:

5.2 Does the software project manager report to a business project manager responsible for the overall benefit of the project to the business?

☐ Yes
☐ No

Comments:

5.3 Does a Software Quality Assurance (SQA) function exist?

☐ Yes
☐ No

If YES, does the SQA function have an independent reporting line from software development project management?

☐ Yes
☐ No

Comments:

5.4 Do management formally assess the benefits, viability, and risk of each software project prior to making contractual (or internal) commitments?

☐ Yes
☐ No

Comments:

(5.5) Are there procedures to ensure that external software subcontracting organisations, if any, follow a disciplined software development process?

☐ Yes
☐ No
☐ N.A.

Applies to: Those organisations which contract out software development to other organisations.

Comments:

5.6 Do management formally conduct periodic reviews of the status of each software project?

☐ Yes
☐ No

Comments:

5.7 For each project, are independent audits (such as inspections or walkthroughs) conducted for each major stage in the software development process?

☐ Yes
☐ No

Comments:

5.8 Is a formal procedure (such as a review or handover with sign-off) used whenever a deliverable (such as a user statement of requirements or system requirements) is passed from one discrete group to another (e.g. *user* to *analyst* to *designer*) to ensure it is properly understood?

☐ Yes
☐ No

Comments:

5.9 Is there a required training programme for all newly-appointed software managers which is designed to familiarise them with in-house software project management procedures?

☐ Yes
☐ No

Comments:

5.10 Is there a procedure for maintaining awareness of the state-of-the-art in CASE or software engineering technology?

☐ Yes
☐ No

Comments:

Section 6 - How Can We Help You?

6.1 Which aspects of your software development activities have the most scope for improvement?

6.2 How can SEA help you achieve those improvements?

Thank you for taking the time to complete the survey. Your co-operation is much appreciated. Please return the survey in the envelope provided, or post to:

SEA
Unit 3, 107 Quay Street,
Brisbane 4000

Qld Jan

Appendix F. Analysis of best practice survey data

Table F.1 Industry sectors of respondent organisations

Sector name	Sector code	Frequency	Percent
Software development	0	156	76.8
Agriculture, forestry & fishing	1	2	1.0
Mining	2	2	1.0
Manufacturing	3	5	2.5
Utilities	4	5	2.5
Construction	5	1	.5
Retail & wholesale	6	1	.5
Accommodation, cafes & restaurants	7	0	0
Transport & storage	8	0	0
Communication services & media	9	2	1.0
Finance & insurance	10	2	1.0
Property & business services	11	1	.5
Government, Administration & Defence	12	5	2.5
Education	13	7	3.4
Health & community service	14	3	1.5
Cultural & recreational services	15	1	.5
Personal & other services	16	0	0
Tourism & hospitality	17	0	0
Information technology	18	10	4.9
Total		203	100.0

Table F.2 Programming languages used now and planned for future

Language	Now		Next 12 months		Trend
	Frequency	%	Frequency	%	%
VB	87	42.9	74	36.5	-6
Ms-Access	74	36.5	57	28.1	-8
Other	68	33.5	33	16.3	-17
C++	64	31.5	54	26.6	-5
SQL	64	31.5	53	26.1	-5
C	47	23.2	32	15.8	-7
Java	35	17.2	53	26.1	9
Delphi	32	15.8	22	10.8	-5
Oracle	31	15.3	24	11.8	-3
COBOL	14	6.9	9	4.4	-2
Powerbuild	10	4.9	4	2.0	-3
PL/I	2	1.0	0	0.0	-1

Note: Languages not used: Telon, Easytrieve & Natural

Table F.3 Other programming languages in use

Programming language or tool	Frequency	Percent
Assembler	6	2.9%
Autodesk Autocad ARX or Lisp	3	1.5%
CA Visual Objects	6	2.9%
Centura	2	1.0%
Clarion	5	2.4%
Clipper	2	1.0%
Fortran	2	1.0%
FoxPro and/or Visual FoxPro	5	2.4%
Informix	4	2.0%
Perl	6	2.9%
RPG	2	1.0%
Paradox	2	1.0%
CA Open Road	2	1.0%
Progress	3	1.5%
Dbase III or IV	3	1.5%
Director Lingo	2	1.0%
MapBasic	2	1.0%
Dataflex	2	1.0%
Pascal	2	1.0%
ABAP/4	2	1.0%
ASP-Jscript	2	1.0%
HTML, MTML, DHTML	3	1.5%
Advanced Revelation and OpenInsight	2	1.0%
MS-Basic & QBasic	2	1.0%
Various un-named	2	1.0%
The following languages were each reported by 1 respondent: 4th Dimension (www.4d.com); Abane IV; Asymetrix Toolbook (www.asymetrix.com); Excel, FilePro; FilemakerPro; Forth (www.forth.com); How; Ingres; Interbase PL/SQL; Javascript; Lotus Notes; Modula Embedded; Paxus RT86; Sybase Power++; Pick; RolePlaying Game (RPG) Director (Macromedia); SQLWindows; TCL?TK; GE Smallworld Magik.		
Note: Respondents were invited to record any other programming languages used now and expected to be used in the next 12 months. Although many respondents chose to record the names of the programming language, they did not consistently tick the check boxes to indicate if usage was current or planned for the future. Therefore, interpretation of the use of these languages in 12 months time is not possible.		

Table F.4 Development tools used now and planned to use in the next 12 months

Tools	Now		Next 12 Months		Trend
	Frequency	%	Frequency	%	
Staff time-sheet system	69	34.0	38	18.7	-15
Source code control	51	25.1	54	26.6	1
3rd party reusable components	49	24.1	35	17.2	-7
Defect tracking	24	11.8	20	9.9	-2
Other	21	10.3	8	3.9	-6
Auto code analysis	12	5.9	13	6.4	0
Automated test capture	5	2.5	11	5.4	3
Code coverage	2	1.0	5	2.5	1

Table F.5 Development tools – recorded comments for 'other' response

Development Tool - Other	Timeframe	Frequency	Percent
No response		180	88.7%
CA visual objects	Now	1	.5
Case Tools (Rat. Rose)	Now	1	.5
Compiling Only	Now	1	.5
Developer 2000	Now	1	.5
Fractal Analysis	Now	1	.5
How	Now & in 12 months	1	.5
In house developed systems ...		1	.5
In house paper-based system	Now	1	.5
In-house software for Pro/Manag	Now	1	.5
Interdev/visual studio	In 12 months	1	.5
Lint	Now & in 12 months	1	.5
Memory checks (eg Purity)	Now	1	.5
Mfch v5-0	Now	1	.5
Ms-Access-Issues register	Now	1	.5
MS Developer Library	Now & in 12 months	1	.5
Own methods used		1	.5
Personal attention	Now	1	.5
Profiling	Now	1	.5
PVCS	In 12 months	1	.5
Rad in magic	Now & in 12 months	1	.5
Source code generation objects	Now	1	.5
Time tracking of tasks	Now	1	.5
User requirements	Now	1	.5
Total		203	100.0

There were 26 responses with comments for 'other' development tools.

Table F.6 Mean adoption level and number of responses for each practice

Question item	Yes	No	Not applicable	Missing	Mean Practice Adoption Level
Section 1—Requirements and Design					
1.1 Is there a procedure for ensuring that appropriate levels of user/ customer/ marketing input is made throughout the project?	169	32	0	2	84.08%
1.2 Is there a procedure to ensure that the systems projects selected for development qualitatively or quantitatively support/alleviate the organisation's business objective/problems?	117	34	47	52	77.48%
1.3 Are design notations such as Structured Analysis and Design Technique used in program design?	101	98	0	4	50.75%
1.4 Are there procedures to ensure that the functionality, strengths, and weaknesses of the 'system' which the software is replacing are formally reviewed?	96	57	46	50	62.75%
1.5 Are prototyping methods used in ensuring the requirements elements of the software?	154	43	0	6	78.17%
1.6 Is a data dictionary available for controlling and storing details of all data files and their fields?	116	55	26	32	67.84%
Section 2—Code and Test					
2.2 Does test planning commence prior to programming beginning based on the user requirements and high-level design documents?	86	111	0	6	43.65%
2.4 Are common coding standards applied to each software project?	157	40	0	6	79.70%
2.5a Is independent testing conducted by users (or appropriate representatives) before any system or enhancement goes live?	162	40	0	1	80.20%
2.5b If YES, is it under the guidance of Software Quality Assurance?	35	118	0	50	22.88%
2.6a Is there a procedure to check that the system configuration (i.e. the programs and any data) passing user acceptance testing is the same as that which is implemented for live operation?	124	74	0	5	62.63%
2.6b Is there a procedure to check that no changes are made directly to a 'live' version of any system (other than through modification to its development version)?	117	74	0	12	61.26%
2.7 Is there a procedure for assuring that regression testing (i.e. the forced re-run of all previous tests prior to any new tests) is routinely performed during and after initial implementation?	53	146	0	4	26.63%

Question item	Yes	No	Not applicable	Missing	Mean Practice Adoption Level
2.8 Are estimates made and compared with actuals for target computer performance (e.g. memory utilisation, processor throughput and file/channel I/O and disk usage)?	47	86	69	70	35.34%
2.9 Are statistics on test efficiency (eg. % of errors actually detected by an activity against the maximum theoretically possible) gathered and analysed for all testing stages in the development process?	21	177	0	5	10.61%
2.10 Are post-implementation software problem reports logged and their resolution effectively tracked and analysed?	153	48	0	2	76.12%
2.11 Do records exist from which (and requiring nothing extra) all current versions and variants of software systems and their components can be quickly and accurately reconstructed in the development environment?	130	69	0	4	65.33%
2.12 Are statistics on the sources of errors in software code gathered and analysed for their cause, detection and avoidance measures?	36	163	0	4	18.09%
2.13 Do procedures exist to ensure that every required function is tested/verified?	107	91	0	5	54.04%
Section 3—Configuration Management					
3.1 Is a change control function established for each software project?	91	109	0	3	45.50%
3.2 Is there a procedure for controlling changes to the software requirements, designs and accompanying documentation?	116	86	0	1	57.43%
3.3 Is there a procedure for controlling changes to the code and specifications?	117	82	0	4	58.79%
3.4 Are software tools used to assist in forwards and/or backwards tracing of software requirements to software designs through to code?	40	160	0	3	20.00%
3.5 Are software tools used for tracking and reporting the status (eg. reviewed, tested, released) of the software/subroutines in the software development library?	39	161	0	3	19.50%
Section 4—Estimates and Schedules					
4.1 Is there a documented procedure for estimating software size (such as 'Lines of Source Code') and thus for using productivity measures?	24	177	0	2	11.94%
4.2 Is a formal procedure used to produce software development effort, schedule, and cost estimates?	77	125	0	1	38.12%
4.3 Are records of actual project resourcing and timescales versus estimates maintained (at individual resource/resource-type level) and regularly analysed/fed-back into the estimating	67	134	0	2	33.33%

Question item	Yes	No	Not applicable	Missing	Mean Practice Adoption Level
and scheduling procedures?					
4.4 Are records of software size maintained for each software configuration item, over time, and fed-back into the estimating process?	24	175	0	4	12.06%
4.5 Is project tracking (e.g. earned value) used throughout the software development process (actual versus planned deliverables analyses, designed, unit tested, system tested, acceptance tested over time) to monitor project progress?	66	134	0	3	33.00%
4.6 Are estimates, schedules and subsequent changes produced only by the project managers who directly control the project resources and are fully aware of their abilities and availabilities?	117	83	0	3	58.50%
4.7 Are software tools used for project planning, estimating, scheduling, and critical path analysis?	87	114	0	2	43.28%
4.8 Does the overall business project manager gain agreement and sign-off from all parties who have produced detailed estimates and schedules before publishing or revising a consolidated project plan?	80	116	0	7	40.82%
4.9 Where other non-software resources are critical to the success of the project is there a procedure for ensuring their availability according to plan?	65	42	92	96	60.75%
Section 5—Project Management & Training					
5.1 Does each software project have a nominated software project manager?	175	26	0	2	87.06%
5.2 Does the software project manager report to a business project manager responsible for the overall benefit of the project to the business?	110	88	0	5	55.56%
5.3 Does a Software Quality Assurance (SQA) function exist?	47	154	0	2	23.38%
5.3 b If 5.3 YES, does the SQA function have an independent reporting line from software development project management?	14	29	0	160	32.56%
5.4 Do management formally assess the benefits, viability, and risk of each software project prior to making contractual (or internal) commitments?	138	63	0	2	68.66%
5.5 Are there procedures to ensure that external software subcontracting organisations, if any, follow a disciplined software development process?	52	42	108	109	55.32%
5.6 Do management formally conduct periodic reviews of the status of each software project?	151	52	0	0	74.38%
5.7 For each project, are independent audits (such as inspections or walkthroughs) conducted	70	132	0	1	34.65%

Question item	Yes	No	Not applicable	Missing	Mean Practice Adoption Level
for each major stage in the software development process?					
5.8 Is a formal procedure (such as a review or handover with sign-off) used whenever a deliverable (such as a user statement of requirements or system requirements) is passed from one discrete group to another (e.g. user to analyst to designer) to ensure it is properly understood?	77	123	0	3	38.50%
5.9 Is there a required training programme for all newly-appointed software managers which is designed to familiarise them with in-house software project management procedures?	42	154	0	7	21.43%
5.10 Is there a procedure for maintaining awareness of the state-of-the-art in CASE or software engineering technology?	42	158	0	3	21.00%
Total of 44 questions	3909	4275	388	748	47.76%
203 responses x 44 questions =8932 responses	43.76%	47.86%	3.78%	8.37%	

Table F.7 Adoption level of each practice – characteristics of distribution

Statistics		Value	Std. Error
Mean		47.7970	3.36379
95% Confidence interval for mean	Lower bound	41.0133	
	Upper bound	54.5808	
5% Trimmed mean		47.7507	
Median		48.1250	
Variance		497.863	
Standard deviation		22.31284	
Minimum		10.61	
Maximum		87.06	
Range		76.45	
Interquartile range		36.5725	
Skewness		.004	.357
Kurtosis		-1.167	.702

Table F.8 Adoption level of each practice - tests of normality

	Kolmogorov-Smirnov(a)			Shapiro-Wilks		
	Statistic	df	*p*	Statistic	df	*p*
Practice Adoption Level	.090	44	.200(*)	.955	44	.083

* This is a lower bound of the true significance.
a Lilliefors Significance Correction

Table F.9 Extent of adoption of each practice ranked in descending order

Practices adopted by 75-100% of organisations	N	%
Each software project has a nominated software project manager	201	87.06
There a procedure for ensuring that appropriate levels of user/customer/marketing input is made throughout the project	201	84.08
Independent testing is conducted by users before any system or enhancement goes live	202	80.20
Common coding standards are applied to each software project	197	79.70
Prototyping methods are used in ensuring the requirements elements of the software	197	78.17
There a procedure to ensure that the systems projects selected for development qualitatively or quantitatively support/alleviate the organisation's business objective/problems	151	77.48
Post-implementation software problem reports are logged and their resolution effectively tracked and analysed	201	76.12
Practices adopted by 50-74.9% of organisations	**N**	**%**
Management formally conducts periodic reviews of the status of each software project	203	74.38
Management formally assesses the benefits, viability, and risk of each software project prior to making contractual (or internal) commitments	201	68.66
A data dictionary available for controlling and storing details of all data files and their fields	171	67.84
Records exist from which (and requiring nothing extra) all current versions and variants of software systems and their components can be quickly and accurately reconstructed in the development environment	199	65.33
Procedures exist to ensure that the functionality, strengths, and weaknesses of the 'system' which the software is replacing are formally reviewed	153	62.75
There is a procedure to check that the system configuration (i.e. the programs and any data) passing user acceptance testing is the same as that which is implemented for live operation	198	62.63
There is a procedure to check that no changes are made directly to a 'live' version of any system (other than through modification to its development version)	191	61.26
Where other non-software resources are critical to the success of the project, there is a procedure for ensuring their availability according to plan	107	60.75
There is a procedure for controlling changes to the code and specifications	199	58.79
Estimates, schedules and subsequent changes are produced only by the project managers who directly control the project resources and are fully aware of their abilities and availabilities	200	58.50
There is a procedure for controlling changes to the software requirements, designs and accompanying documentation	202	57.43
The software project manager reports to a business project manager responsible for the overall benefit of the project to the business	198	55.56
There are procedures to ensure that external software subcontracting organisations, if any, follow a disciplined software development process	94	55.32
Procedures exist to ensure that every required function is tested/verified	198	54.04
Design notations such as Structured Analysis and Design Technique are used in program design	199	50.75

Practices adopted by 25-50% of organisations	N	%
A change control function is established for each software project	200	45.50
Test planning commences prior to programming beginning based on the user requirements and high-level design documents	197	43.65
Software tools used for project planning, estimating, scheduling, and critical path analysis	201	43.28
The overall business project manager gains agreement and sign-off from all parties who have produced detailed estimates and schedules before publishing or revising a consolidated project plan	196	40.82
A formal procedure (such as a review or handover with sign-off) is used whenever a deliverable (such as a user statement of requirements or system requirements) is passed from one discrete group to another (e.g. *user* to *analyst* to *designer*) to ensure it is properly understood	200	38.50
A formal procedure is used to produce software development effort, schedule, and cost estimates	202	38.12
Estimates are made and compared with actuals for target computer performance (memory utilisation, processor throughput and file/channel I/O and disk usage)	133	35.34
For each project, independent audits (such as inspections or walkthroughs) are conducted for each major stage in the software development process	202	34.65
Records of actual project resourcing and timescales versus estimates maintained (at individual resource/resource-type level) and regularly analysed/fed-back into the estimating and scheduling procedures	201	33.33
Project tracking (e.g. earned value) is used throughout the software development process (actual versus planned deliverables analyses, designed, unit tested, system tested, acceptance tested over time) to monitor project progress	200	33.00
The SQA function has an independent reporting line from software development project management	43	32.56
There is a procedure for assuring that regression testing (i.e. the forced re-run of all previous tests prior to any new tests) is routinely performed during and after initial implementation	199	26.63
Practices adopted by 0-25% of organisations	**N**	**%**
A Software Quality Assurance (SQA) function exists	201	23.38
Independent testing is under the guidance of Software Quality Assurance	153	22.88
There is a required training programme for all newly-appointed software managers which is designed to familiarise them with in-house software project management procedures	196	21.43
There a procedure for maintaining awareness of the state-of-the-art in CASE or software engineering technology	200	21.00
Software tools used to assist in forwards and/or backwards tracing of software requirements to software designs through to code	200	20.00
Software tools are used for tracking and reporting the status (reviewed, tested, released) of the software/subroutines in the software development library	200	19.50
Statistics on the sources of errors in software code gathered and analysed for their cause, detection and avoidance measures	199	18.09
Records of software size are maintained for each software configuration item, over time, and fed-back into the estimating process	199	12.06
There is a documented procedure for estimating software size (such as ‘Lines of Source Code’) and thus for using productivity measures	201	11.94
Statistics on test efficiency (eg. % of errors actually detected by an activity against the maximum theoretically possible) are gathered and analysed for all testing stages in the development process	198	10.61

Table F.10 Comparison of adoption of primary life cycle practices compared to organisation/support practices

Life cycle group	N	Mean	Std. deviation	Std. error mean
Primary	13	56.7062	24.94557	6.91866
Organisation/support	31	44.0610	20.39296	3.66268

Table F.11 Adoption of primary life cycle practices compared to support/ organisational practices - independent samples test

Levene's test for equality of variances		t-test for equality of means						
F	*p*	t	df	*p* (2-tailed)	Mean difference	Std. error difference	95% confidence interval of the difference	
Equal variances assumed							Lower	Upper
.739	.395	1.756	42	.086	12.6452	7.20030	-1.88561	27.17598
Equal variances not assumed								
		1.615	19.07	.123	12.6452	7.82835	-3.73569	29.02607

Table F.12 Organisation adoption level - skewness and kurtosis

Developer Group	Skewness	Kurtosis
Does not develop COTS	.18	-.55
COTS software developer	.11	-.89
Note: values for skewness and kurtosis are zero if the observed distribution is exactly normal.		

Table F.13 Comparison adoption level of COTS and non-COTS developers - test of homogeneity of variance

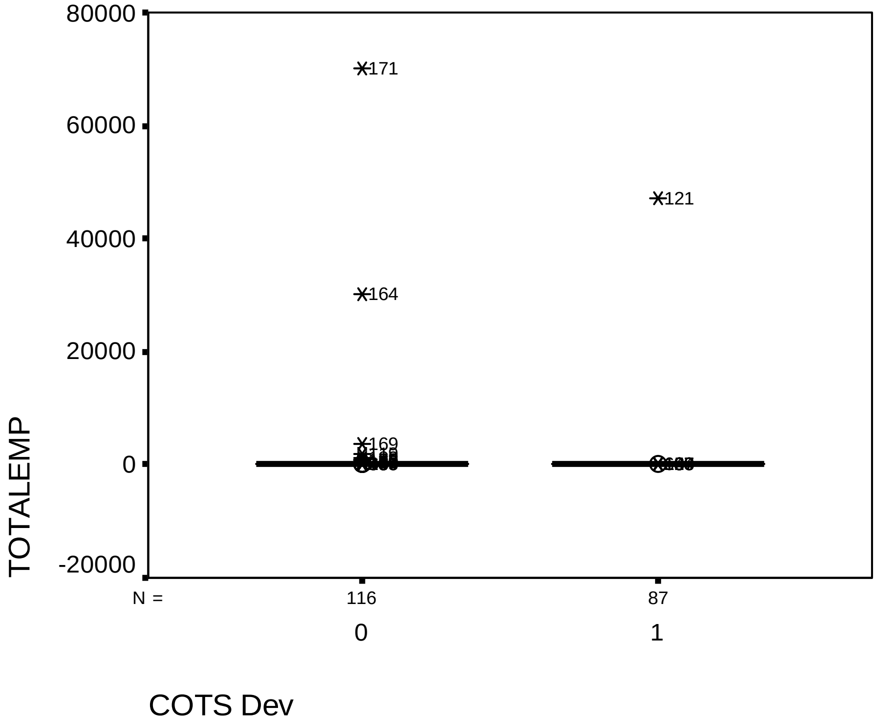
Adoption level	Levene statistic	df1	df2	*p*
Based on mean	.493	1	201	.483
Based on median	.478	1	201	.490
Based on median and with adjusted df	.478	1	200.827	.490
Based on trimmed mean	.509	1	201	.477

Table F.14 Adoption level of COTS developers compared to non-COTS developers

COTS developer	N	Mean	Std. deviation	Std. error mean
No-0	116	44.1789	20.18038	1.87370
Yes-1	87	52.1740	21.33042	2.28686

Table F.15 Independent samples test comparing adoption level of COTS with non-COTS developers

Levene's test for equality of variances		t-test for equality of means						
F	*p*	t	df	*p* (2-tail)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
Equal variances assumed							Lower	Upper
.493	.483	-2.726	201	.007	-7.9952	2.93302	-13.77860	-2.21172
Equal variances not assumed								
		-2.704	179.67	.008	-7.9952	2.95643	-13.82895	-2.16136



Non-COTS COTS

Figure F-1 Box plot comparison of number of employees for non-COTS and COTS developers

Table F.16 Correlation: total number of employees and adoption level

Statistic	All responses	Non-COTS total employees			COTS total employees
	All cases	All cases	Excluding 2 outliers	All cases	Excluding 1 outlier
Pearson correlation	.060	.060	.112	.060	.254(**)
p (1-tailed)	.262	.262	.117	.292	.009
N	116	116	114	87	86

** Correlation is significant at the 0.01 level (1-tailed).

Table F.17 Correlation: adoption level and total number of employees excluding software users (in-house or 3rd party)

	Non-COTS total employees	COTS total employees
Pearson correlation	.211(*)	.337(**)
p (1-tailed)	.042	.002
N	68	74

* Correlation is significant at the 0.05 level (1-tailed).
** Correlation is significant at the 0.01 level (1-tailed).

Table F.18 Correlation: adoption level and total number of employees for software users (in-house or 3rd party)

	Non COTS total employees	COTS total employees
Pearson correlation	.119	.086
p (1-tailed)	.210	.390
N	48	13

Table F.19 Correlation: adoption level and number of software developers

	Non-COTS number of developers	COTS number of developers
Pearson correlation	.253(**)	.302(**)
p (1-tailed)	.003	.002
N	116	87

** Correlation is significant at the 0.01 level (1-tailed).

Table F.20 Adoption by sector comparison of non-COTS and COTS developers

Sector name	Sector code	Total	Not COTS Developer		COTS Developers	
		N	N	Mean adoption	N	Mean adoption
Software development	0	133	78	45.7%	78	52.9%
Agriculture, forestry & fishing	1	2	1	73.8%	1	51.2%
Mining	2	2	2	22.3%		
Manufacturing	3	5	3	42.5%	2	27.3%
Utilities	4	5	4	58.0%	1	35.7%
Construction	5	1			1	15.8%
Retail & wholesale	6	1	1	28.6%		
Accommodation, cafes, restaurants	7	0				
Transport & storage	8	0				
Communication services & media	9	2	2	27.6%		
Finance & insurance	10	2	2	64.2%		
Property & business services	11	1	1	50.0%		
Government admin & defence	12	5	5	50.0%		
Education	13	7	5	16.1%	2	62.0%
Health & community service	14	3	2	36.8%	1	63.4%
Cultural & recreational services	15	1	1	61.4%		
Personal & other services	16	0				
Tourism & hospitality	17	0				
Information technology	18	10	9	39.1%	1	70.7%
Total		203	116		87	

Table F.21 Non-COTS Developers: adoption level by sector
Test of homogeneity of variances

Levene Statistic	df1	df2	p
1.363(a)	9	102	.215

a Groups with only one case are ignored in computing the test of homogeneity of variance for adoption level (4 sectors were ignored).

Table F.22 Non-COTS Developers: Comparison of adoption levels by sector

One-Way ANOVA	Sum of squares	df	Mean square	F	*p*
Between groups	9174.639	13	705.741	1.912	.037
Within groups	37658.858	102	369.204		
Total	46833.497	115			

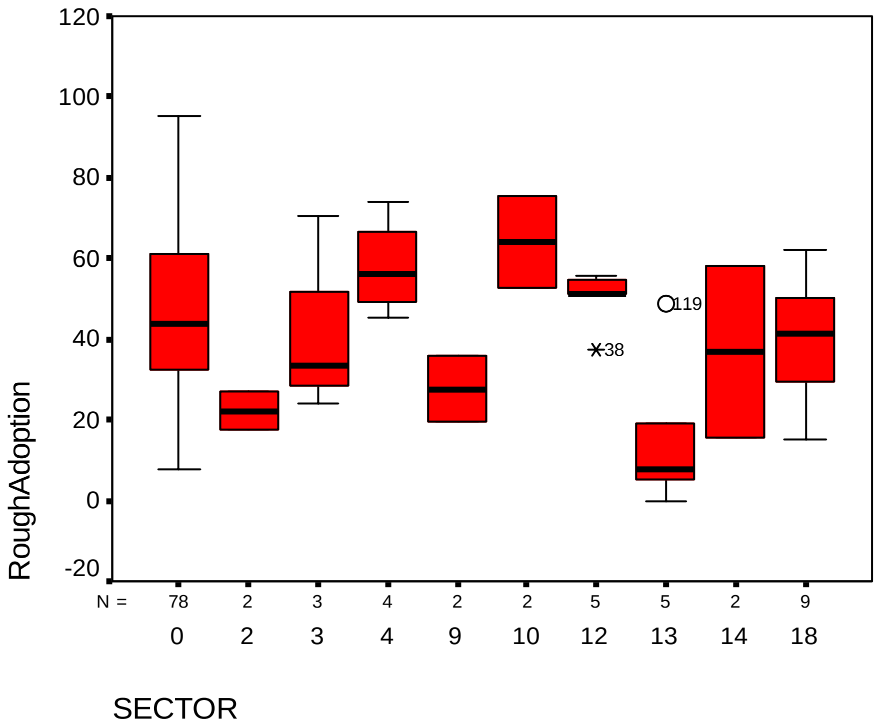


Figure F-2 Boxplot showing Comparison of mean adoption by sector for non-COTS

Table F.23 Non-COTS developers: summary of significant differences from multiple comparisons

Sectors Compared	Mean Difference	Std. Error	*p*	95% Confidence Interval	
				Lower Bound	Upper Bound
Software development-Education	29.6583(*)	8.86421	.001	12.0762	47.2405
Mining-Utilities	-35.6973(*)	16.64041	.034	-68.7035	-2.6911
Mining-Finance & insurance	-41.9303(*)	19.21469	.031	-80.0425	-3.8180
Utilities – Education	41.8840(*)	12.88961	.002	16.3175	67.4504
Finance & insurance – Education	48.1169(*)	16.07617	.003	16.2299	80.0039
Govt admin & defence-Education	33.9133(*)	12.15244	.006	9.8090	58.0176
Education –IT	-23.0583(*)	10.71744	.034	-44.3163	-1.8003

* The mean difference is significant at the .05 level.

Table F.24 COTS Developers: adoption level by sector
Test of homogeneity of variances

Levene statistic	df1	df2	*p*
1.758(a)	2	79	.179

a Groups with only one case are ignored in computing the test of homogeneity of variance for Adoption. (5 sectors ignored)

Table F.25 COTS Developer: Adoption level by sector - comparison of means
One-way ANOVA - Adoption level

	Sum of squares	df	Mean square	F	*p*
Between groups	3533.168	7	504.738	1.120	.359
Within groups	35595.714	79	450.579		
Total	39128.882	86			

Appendix G. PIP Sample Documents

Section 1 Organisation Context Questionnaire

SOFTWARE QUALITY INSTITUTE

PROCESS IMPROVEMENT PROJECT

ORGANIZATION CONTEXT INFORMATION

1. Company Data

Company Name:

Department:

Address:

Contact Person:

Phone:

Fax:

Email:

Year of Foundation:

2. Employees

Number of Employees:

Full Time: ____

Pat Time: ____

Contract: ____

Total: ____

Education:

Postgraduate: ____

Graduate: ____

Other: ____

Function:

Technical: ____
Support/Admin: ____

Experience:

More than 5 years: ____
Less than 5 years: ____

3. Market Segment

Application Domain:

For which business sector does your organization deliver/acquire software? (please select all that apply)

- ❑ Finance (excluding banking)
- ❑ Insurance
- ❑ Banking
- ❑ Petroleum
- ❑ Automotive
- ❑ Public Utilities (Gas, Water, Electricity)
- ❑ Aerospace
- ❑ Telecommunications
- ❑ Public Administration
- ❑ Consumer Goods
- ❑ Retail
- ❑ Defence
- ❑ Information Technology/Software
- ❑ Health and Pharmaceutical
- ❑ Leisure and Tourism
- ❑ Manufacturing
- ❑ Construction
- ❑ Travel
- ❑ Media (Television, Radio)
- ❑ Education
- ❑ Distribution/Logistics
- ❑ Other (*Please Specify*) ------------------------------------

Applied Technologies:

- ❑ OO
- ❑ 4GL
- ❑ Scripting
- ❑ Real Time
- ❑ Internet
- ❑ AI
- ❑ Client Server
- ❑ Other (*Please Specify*) ____________________

4. Projects

No. of projects as Work in Progress: ____________________
Typical no. of employees per project: ____________________
Typical duration to final delivery: ____________________
Average cost overrun (as a percentage); ____________________

5. ISO 9001

Does your organisation have ISO9001 certification?

Please circle: Yes/No

If yes, on which date was the ISO 9001 certification awarded?

(mm/yyyy) __/____

6. ISO 9001

How would you best **characterize** your organization's … *(please select one in each row)*

	Excellent	Good	Fair	Poor	Don't Know
Ability to meet budget commitments?					
Ability to meet schedule commitments?					
Ability to achieve high customer satisfaction?					
Ability to meet specified and implied requirements?					
Staff productivity?					
Staff morale/job satisfaction?					

How **important** are the following performance measures to your organization? *(please select one in each row)*

	Very Important	Important	Somewhat Important	Not Important
Ability to meet budget commitments?				
Ability to meet schedule commitments?				
Ability to achieve high customer satisfaction?				
Ability to meet specified and implied requirements?				
Staff productivity?				
Staff morale/job satisfaction?				

Thank you for taking the time to complete this questionnaire.

Please return this questionnaire by email to sqi@sqi.gu.edu.au or alternatively we will collect it during your assessment.

Section 2 Sample of assessment template (requirements elicitation)

Requirements gathering: How do you gather requirements from customers and potential users?

PA 1.1 - Process Performance: Is the requirements gathering process performed, at least informally?		
Does the process for gathering requirements as implemented in this organization achieve its expected outcomes?	Refer specific outcomes on next page	
Do the company personnel understand the scope of the requirements gathering process?		
Are there identifiable input work products for requirements gathering? Identify.	List identified inputs	
Are there identifiable output work products from requirements gathering? Identify.	List identified outputs	
Comments		

Has your company established continuing communications with your key customers?	Newsletter Web Page User Group	
Do you have a clear understanding of the customer's requirements for each project?	Functions Quality Characteristics Safety Security	
Is there a means for identifying new customer needs and reflecting this in the requirements?	Environmental scanning Changes in business domain	
Do you monitor the needs of your customers on a continuous basis?	Surveys Market research Customer satisfaction	
Can your customers readily establish the status of their requests?	Web page	
Do you have a program for ongoing enhancement of your products?	Release Policy	

PA 2.1 - Performance Management: Is the performance of the requirements gathering process managed?		
Do you identify your objectives for requirements gathering?	Quality Cost Schedule	
Do you develop a plan for your requirements gathering activities?	Scope Schedule Work breakdown	
Do you assign specific responsibilities and authorities for developing work products associated with requirements gathering?	Roles Responsible individuals Approvals	
Do you track and monitor the requirements gathering activities, and re-plan when needed?	Status reports Team meetings	
Comments:		

PA 2.2 – Work Product Management: Are the work products of the requirements gathering process managed?		
Do you specify requirements for the work products associated with requirements gathering?	Documentation Standards Control requirements Timing	
Do you manage the documentation and change control for the work products associated with requirements gathering?	Version control Baseline definition	
Have you identified and defined any dependencies between the work products associated with requirements gathering?	Relationships between work products Sequencing	
Do you evaluate and where necessary take corrective action to ensure the quality of the work products associated with requirements gathering?	Reviews Records of defects found Traceability of correction	
Comments		

PA 3.1 - Process Definition: Is the requirements gathering process defined?		
Do you have a standard process for requirements gathering, and does it provides guidance on implementation and tailoring?	Policies Procedures Standards	
Do you implement or tailor the standard process for requirements gathering to obtain a defined process appropriate to the project or product?	Common approach to tailoring for individual projects	
Do you collect performance data about requirements gathering so that the behaviour of your defined process can be understood?	Suitability of measures	
Do you establish and refine your understanding of the behaviour of the requirements gathering process by using relevant performance data?	Use of measures	
Do you refine the standard process for requirements gathering?	Improvement suggestions Change requests	
Comments		

PA 3.2 - Process Resource: Are suitable resources provided for the requirements gathering process?		
Do you identify and document the roles, responsibilities and competencies required to support the implementation of your defined process for requirements gathering?	Job descriptions Skills matrix	
Do you provide the human resources needed to support the performance of the defined process for requirements gathering? Are these resources allocated and used?	Training needs Training records Recruitment policy	
Do you identify and document the infrastructure requirements to support the implementation of the defined process for requirements gathering?	Is the infrastructure adequate Tools Environment	
Is the identified process infrastructure provided to support the performance of the defined requirements gathering process? Is this infrastructure allocated and used?	Upgrading development tools Hardware improvements Furnishings	
Comments		

Section 3 List of questions: generic and base practice questions

Generic Practice Questions (these are tailored and used for each process):

PA 1.1 - Process Performance: Is the requirements gathering process performed, at least informally?
Does the process for gathering requirements as implemented in this organization achieve its expected outcomes? Do the company personnel understand the scope of the requirements gathering process? Are there identifiable input work products for requirements gathering? - Identify Are there identifiable output work products from requirements gathering? – Identify
PA 2.1 - Performance Management: Is the performance of the requirements gathering process managed?
Do you identify your objectives for requirements gathering? Do you develop a plan for your requirements gathering activities? Do you assign specific responsibilities and authorities for developing work products associated with requirements gathering? Do you track and monitor the requirements gathering activities, and re-plan when needed?
PA 2.2 – Work Product Management: Are the work products of the requirements gathering process managed?
Do you specify requirements for the work products associated with requirements gathering? Do you manage the documentation and change control for the work products associated with requirements gathering? Have you identified and defined any dependencies between the work products associated with requirements gathering? Do you evaluate and where necessary take corrective action to ensure the quality of the work products associated with requirements gathering?
PA 3.1 - Process Definition: Is the requirements gathering process defined?
Do you have a standard process for requirements gathering, and does it provides guidance on implementation and tailoring? Do you implement or tailor the standard process for requirements gathering to obtain a defined process appropriate to the project or product? Do you collect performance data about requirements gathering so that the behaviour of your defined process can be understood? Do you establish and refine your understanding of the behaviour of the requirements gathering process by using relevant performance data? Do you refine the standard process for requirements gathering?
PA 3.2 - Process Resource: Are suitable resources provided for the requirements gathering process?
Do you identify and document the roles, responsibilities and competencies required to support the implementation of your defined process for requirements gathering? Do you provide the human resources needed to support the performance of the defined process for requirements gathering? Are these resources allocated and used? Do you identify and document the infrastructure requirements to support the implementation of the defined process for requirements gathering? Is the identified process infrastructure provided to support the performance of the defined requirements gathering process? Is this infrastructure allocated and used?

Base Practice Questions

The following questions are included with the PA 1.1 - Process Performance questions and are specific to the particular process

Requirements elicitation:
Has your company established continuing communications with your key customers? Do you have a clear understanding of the customer's requirements for each project? Is there a means for identifying new customer needs and reflecting this in the requirements? Do you monitor the needs of your customers on a continuous basis? Can your customers readily establish the status of their requests? Do you have a program for ongoing enhancement of your products?
Software development
In the course of developing your products do you generate intermediate products such as: Requirements specification; Design description; Report on integration testing; System test report; Other(s)? Do you take steps in software development to ensure that all of the requirements are addressed in the design? Can you demonstrate from results (such as from testing) that the product developed meets the initial requirements? Do you have a mechanism for formal acceptance of the product by the customer?
Configuration management
Do you have a strategy for configuration management? Are all items generated by your project identified, defined and baselined? Are there controls in place for modifications and releases of your products? Do you record and report on the status of the individual products and requests for modification? Do you ensure that modifications and new releases to your product are complete and consistent? Do you control the storage, handling and delivery of the modifications and new releases to your product?
Quality Assurance
Have you developed and implemented a strategy for performing software quality assurance? Do you record and store evidence of your software quality assurance activities? Do you identify problems or non-conformances with contract requirements? Can you demonstrate that your software products processes and activities conform to relevant standards, procedures and requirements?
Problem resolution
Do you have a defined way to ensure that all discovered problems are analysed and resolved? Do you prepare problem reports upon detection of problems (including non-conformances) in a software product or activity? Do you have a mechanism for recognizing and acting on trends in problems identified?
Project management
Do you define and document the scope of the work for the project, over both requirements gathering and software development? Do you evaluate the feasibility of achieving the goals of the project within available resources and constraints? Have the tasks and resources necessary to complete the work been sized and estimated? Have the interfaces between the project, and other projects and departments, been identified and are they monitored? Have plans for the project been developed and implemented? Has the progress of the project been monitored and reported? Do you take actions to correct deviations from the plan and to prevent recurrence of problems identified in the project, when targets are not achieved?

Risk Management
Have you defined and determined the scope of the risk management to be performed for the project? Have you defined and implemented appropriate risk management strategies? Are risks to the project identified in the project plan, and as they develop during the conduct of the project? Do you analyse risks and use the results to prioritise the resources to monitor these risks? Have you defined measures of risk, and then applied these to assess the risk status of your project? Do you take actions to correct or avoid the impact of risk?
Process establishment
Do you have a standard set of policies and methods for projects approved for use in your company? Are the plans and procedures for your projects based upon these standards? Do your standards include descriptions of the common tasks and activities to be followed in every project? Do you modify the standard methods for each project you undertake? Do you retain and use documents and data from previous projects conducted in the organisation?

Section 4 Questions on PIP feedback forms

Feedback: about the assessors

11 forms analysed (2 types of response forms), frequencies calculated.
The first 7 questions were answered on a 4 point Likert scale (almost always, more often than not, sometimes, rarely if ever):
Was it clear why the information was requested during the assessment?
In your judgement, was the information provided by the assessees during the assessment within the scope of the assessment?
Were you concerned during the assessment about possible breaches of confidentiality by the assessors?
Did the assessors appear to have any biases during the assessment?
Did the assessors demonstrate understanding of the processes being assessed?
Did the assessors demonstrate an adequate understanding of the OU and its business?
In your judgement, did the assessors behave in a professional manner during the assessment? 4 point Likert scale: excellent, good, fair, poor).
In your judgement, how would you characterize the competence of the assessors who conducted the assessment?

The following questions were only on type A forms (release date 23 June 1999)
4 responses analysed
Did you verify the competence of the assessor? (yes/no)
How was competence verified? (4 options)

The following questions were only on type B forms - eg. SQI-PIP-RA13-FB ver 0.3 2 Dec 1999
6 responses analysed
How closely did the assessment meet your expectations? (4 point Likert scale: totally, largely, partially, not at all)
To what extent did the final report reflect the understanding reached at the site visit? (4 point Likert scale: totally, largely, partially, not at all)
How closely do the findings from this assessment reflect your own understanding of your organization's capabilities? (4 point Likert scale: totally, largely, partially, not at all)
Comments

Value of assessment

Overall, how would you characterize your understanding of the assessment process and its results? (Excellent, good fair, poor)
Was the process profile produced by the assessment clearly stated and easy to understand? (5 point Likert scale)
To what extent did you understand the purpose of the activities that took place as part of the assessment? (5 point Likert scale)
To the best of your knowledge, within the scope of the assessment, how accurately did the process profile indicate the OU's major problems? (3 point Likert scale)
Did the process fail to identify any problems within the scope of the assessment? (yes/no)
Did the process wrongly identify anything as a problem? (Yes/no)
To the best of your knowledge, how well did the assessment results characterize the OU's strong points? (3 point Likert scale)

Do you believe that the assessment was worth the expense and time expended? (3 point Likert scale)

Statements about the usefulness of the assessment. 5 point Likert scale (strongly agree, agree, disagree, strongly disagree, don't know).
The assessment provided valuable direction about priorities for process improvement within the OU.
The assessment helped us better understand what needs to be improved.
We still need more guidance about how to go about process improvement
The assessment improved awareness, buy-in, and support for PI among the technical staff in the OU
The assessment was impractical; it took too long and cost too much
The wrong people or projects were selected to participate in the assessment (e.g. people that were interviewed or those who filled up questionnaires)
People weren't fully honest with the assessment team
It was easy to understand the processes followed by the OU in terms of the Assessment Model that was used
The assessment Model that was used provides real direction for long-term software process improvement
There are important areas that the Assessment Model that was used does not address.

Appendix H. Summary of PIP assessment and final reports

Firm No.1
Summary of strengths/weaknesses at initial assessment
Firm #1 showed higher capabilities in the areas of customer requirements gathering and the management of risk, with the software development process maturing as the product matured. All the processes evaluated were performed by competent and effective staff. The Managing Director was actively involved with the day-to-day running of the business. The software design for one of the software products was stable and ready to be documented, thereby reducing the risk of further maintenance activities. An adaptation of the Evolutionary Rapid Development process was in use and a check-list could provide more visibility to management. The configuration management activities could benefit from a support tool, and code analysers could help improve the consistency of style, commenting and layout of programs.

In order to promote standardisation for new developers, standards for style, comments, and layout for each language was recommended. Assessors noted the need for a classification system and bug-tracking tool for classing and recording the severity and priority of problems and bugs. A work break down structure and schedule for the next project could incorporate lessons learned, and provide visibility of progress to all, including management. It was suggested that the software development sample in MS Project could be adapted to the needs of Firm #1.

Summary of follow-up meeting
Since the initial assessment, the company sold the distribution rights to its core software to another company and consequently, the number of full time employees had dropped to two, and the focus of firm #1 had changed to technology development, particularly in the security systems area. The focus changed to defining and developing technology demonstrators that can be packaged and marketed internationally.

The assessment helped the business owner to prioritise the business goals and reassess the business direction, in particular addressing staff issues. Two of the recommendations were implemented, the creation of a baseline plan, and the introduction of configuration management tools.

The change in focus to research and develop new technologies for the company product range had altered the priorities for the Owner. Based on his personal international contacts, he was actively pursuing new opportunities in the access security and control domain.
[Information summarised and adapted from SQI-PIP-FR01-01.doc]

Firm No.2
Summary of strengths/weaknesses
Firm #2 had developed strong relationships with its clients, and documents the results of client discussions to arrive at concrete and agreed requirements for its projects. The process of software development was reasonably documented with good records of system design. A strong emphasis was placed on portability and potential for reuse in the designs and implementation. Firm #2 provided total support for their clients, adopting a ‘whole system’ support approach that entrenches

client loyalty. The company had a reputation for on-time delivery of products that generally meet the client's expectations, and managed the projects to maintain this reputation, and invested strongly in identification and mitigation of risks associated with the development activities.

However, there was no formal system for configuration management with a resulting threat to ongoing product integrity. Existing manual systems were largely enforced through professional discipline and could prove difficult to maintain as the company grows. Records of product validation, through internal testing and client-based beta testing, were limited and did not provide adequate assurance of product quality.

The existing program of informal reviews was not documented and little 'corporate knowledge' was captured in any documented form. Furthermore, records of process performance were limited, resulting in problems with project estimation and risk monitoring. Also, no mechanisms existed to ensure that the existing (informal) high standards in place in the organization were adopted by contractors or potential new staff. The assessors also noted that there was no formal process in place to ensure that the current 'good intentions' to improve the existing set of policies and procedures were followed up in a timely and effective manner.

Summary of follow-up meeting
The follow-up assessor's efforts to contact Firm #2 by telephone and email during August 2000 were unsuccessful. Firm #2 was no longer registered in local telephone directories or on the Software Engineering Australia contact directories. It was therefore concluded that Firm #2 had ceased to operate.
[Information summarised and adapted from SQI-PIP-FR02-01 PAT.doc]

Firm No.3
Summary of strengths/weaknesses
Firm#3 had a strategy to build, market and maintain software products to industries that operate and manage capital-intensive assets. The company employed competent, experienced people who recognised the need for process development alongside product development, but the lack of available additional competent people had limited the rate of development. This had exacerbated the need for the company to invest in sound processes, as contractors may be required for future work.

At firm #3, requirements' gathering was well developed; excellent interaction with clients to ensured that the needs of the market were well understood. The development process was well documented, with evidence to confirm that the documented requirements had been implemented in the delivered product. Generally, configuration management was well practiced ensuring high confidence in the integrity of the delivered software. A sound process for recording and resolving problems was in place, with good monitoring of progress. A detailed quality manual had been produced providing sound policies and procedures for many aspects of software development activities.

However, it was noted that documentation was incomplete in some areas in software development, notably in the definition and execution of testing. No formal approach existed for estimating the size of the work to be performed. Tasks outside of fundamental development activities were not identified and scheduled as part of the planning process. While risks are identified at the beginning of a project, no mechanism existed to monitor them on an ongoing basis. Although the basic elements of a quality assurance system were defined, the system had not been effectively implemented. No regular reports were produced for either monitoring or performance enhancement purposes for either configuration management or problem resolution. No mechanism was in place to ensure the structured development and implementation of an effective set of process assets.

Summary of follow-up meeting

Since the assessment, firm #3 had adopted a new development methodology that allowed a more appropriate model to be used throughout the development environment, rather than adhering to traditional Waterfall model. This had enabled firm #3 to introduce more considered approach to software design and advanced project management techniques. Work had also commenced on the adoption of a more thorough configuration management practice, which could be tailored to individual site installations.

No re-evaluation of any of the processes was performed during the follow-up contact, as the interviewee indicated that the newly established processes were immature at this stage. Firm #3 confirmed that the Assessment afforded considerable value to their company, not only in providing an objective opinion of their current development environment, but in providing legitimatisation of the established development activities. The SQI’s role in the assessment was also viewed as providing an initial point for creating a ‘mentoring’ role in implementing selected improvement activities.

New procedures had been put in place to formalise the documentation of testing activities. Consideration had been given to establishing checklists and other methods to assist the validation of the requirements gathering process. A more detailed WBS was in use by the project manager for resourcing and task allocation, also recording all task efforts. New configuration management practices were adopted to monitor all items of development work in progress, and control and manage them through to testing and release. Basic templates have been established to enable standardisation across the environment. New practices for problem management were in place to enable problems to be raised, prioritised, traced, and consequently managed through formal change request forms.
[Information summarised and adapted from SQI- PIP-FR03-02.doc]

Firm No.4

Summary of strengths/weaknesses

In the context of the initial development of its principal software product, firm #4 demonstrated good capabilities in most processes, particularly configuration management and problem resolution, with risk management and process establishment the only weak processes identified. The key strength of the company was the in-depth experience and knowledge of the development team. Once a customer base was established, potential risks could arise, particularly in configuration management and risk management. Opportunity existed to establish the management and definition of the processes assessed before acquiring the customer base. This would enable firm #4 to grow, whilst maintaining and improving its current process capabilities demonstrated during the initial development phase.

The most striking factor for this organisation was the strong management control applied to the project. With limited resources, and a lot of will-power a product had been developed that was ready to market. Critical work products such as work breakdown structure, milestones and schedules, problem recording and reporting, configuration management files, requirements traceability tables and user testing were all evident. The adaptation of the Booch method for OO development was also noteworthy.

Summary of follow-up meeting
No actions have been reported for Firm #4 since the Assessment had been performed. No re-evaluation of process capability had been performed. The assessment sponsor had reduced his involvement in the firm due to family illness.
[Information summarised and adapted from SQI-PIP-FR04-01pat.doc]

Firm No.5
Summary of Strengths/Weaknesses
Firm #5 had a generally informal process for development of software that was supported by excellent tool selection, leading to high confidence in the integrity of the delivered software. The firm placed significance on the service and support of its customer base. Considerable effort had been invested in the development of user requirements for the core product.

The informality in the development processes was seen as constituting significant risks for the firm in an environment of system and market growth. There was a need to adopt a more formal and structured approach to both technical and management issues. It was recommended that more attention should be paid to aspects of software development, in particular testing, and also to the establishment of a coherent strategy for project management.

Summary of follow-up meeting
Since the assessment, firm #5 had relocated to new offices, and the staffing profile had changed slightly, with additional domain expertise and fewer part-time staff. The firm had been reorganized, with the aim of reducing the managerial load on the senior manager. There had been noticeable growth in business opportunities, with a major contract under negotiation.

The development process had been formalised. Project plans, containing a detailed statement of scope for the work to be performed, were now produced for all work, which was still monitored using the requests and defects system. A specification of requirements, based upon IEEE Std 830, had been introduced. The requests and defects system had been improved and was now used as a key driver for all work in the firm. Formal projects were linked to existing requests, and corrective maintenance was managed using the requests and defects system.

New procedures had been introduced in relation to the control of report generation routines, where a major problem with consistency and integrity had been found. The range of application of the configuration management system had been expanded, partly in response to this problem. Individual projects were now defined and the scope of work was clearly documented. The project plan supported better decisions

on feasibility, which was reinforced by more effective contract reviews. The firm had documented and disseminated a quality policy, and a variety of quality records were now identified and retained.
Risks were now routinely identified for all projects, and mitigation strategies were defined. There had been significant activity in the development of new and revised procedures for software development and project management. However, the process for establishing these additional process assets remained largely ad-hoc and uncontrolled.

With increased awareness of the importance of measurement, a number of relevant data items were now collected on a routine basis, and some of the key systems in the firm, including the requests and defects system, have been modified to improve data collection. A timesheet system had been introduced for recording effort, with work breakdown codes in use. Although there was much more data being collected, there was very limited analysis of the data, and the impact of the added data on actual project performance was minimal.
[Information summarised and adapted from SQI-PIP-FR05-10.doc]

Firm No.7
Summary of Strengths/Weaknesses
At Firm #7, the strong project management focus and a technically competent workforce were identified as major strengths. The business needs and the desire to maintain a good reputation for software and software development drove the risk management process effectively. Requirements gathering, software development, configuration management, and process establishment were sound, but have areas where improvements would be possible. The assessors noted that the directors searched out good ideas and adapted them to their own environment, for example, the use of the material from NASA, and the IEEE.

At firm #7, it was recognised that the importance of the quality assurance process would increase as additional staff or contractors were hired. The need to identify the standards to be used, and verify that they have been applied effectively was a critical factor in ISO 9001 certification. It was also labour intensive. The problem resolution process and the issue management system were labour intensive and needed an effective automated system to facilitate analysis of trends. It was noted that the demands of one major client had the potential to impact on the business goals of firm #7, resulting in the unavailability of critical resources for product development. Also, if management's attention was diverted to process improvement activities, then revenue-generating activities may suffer. Rather than introduce improvement targets to improve productivity, the assessors considered it more helpful to focus on training, infrastructure support, and effective tools.

Summary of follow-up meeting
Since the initial assessment, firm #7 had relocated to larger premises. Major development work was planned for an upgraded user interface, to provide a consistent look and feel for the product. The company had consolidated its business and was extending the product to a wider, more generic market. Also, opportunities in the Defence market had been identified. No changes to any of the target processes were identified at the follow-up meeting. However, a repository of measures to be collected from projects had been identified and would be incorporated in the new

product. Firm #7 had hoped to participate in the SEA showcase program, and when that that did not eventuate, firm #7 addressed the process improvement process on an ad hoc basis, and this was interrupted by the relocation.
[Information summarised and adapted from SQI-PIP-FR07-01.doc]

Firm No.8
Summary of Strengths/Weaknesses
The assessment identified strengths in the software development capability of firm #8 in the areas of risk management, quality assurance and quality management, focus on customer satisfaction, project management, and the use of CASE tools. However, loss of key personnel in a very competitive market was seen as a constant risk. Human resource management was a key issue for the company.

There appeared to be incomplete processes and shortcuts in the development process: it was very easy to do a code-compile-test cycle without placing the code under configuration control. Unless the configuration management system was integrated with the CASE tool, and used consistently, development would become difficult to manage. Definition and formalisation of validation and verification processes (especially time estimation and resource allocation) could help reduce the costs of service and on-site maintenance. A project characteristics profile could help to identify which processes are most critical for specific projects.

Incorporation of a measurement plan in the project plan could help management get better visibility of performance, and progress, and make decisions about the project in time to influence the outcome. Maintaining the mentoring program for developers, and extending it to project managers could provide incentives to reduce staff turnover. Configuration control using tools can be used for software code and documentation, but are most effective when integrated with the CASE tools. Also, an effective problem tracking system would provide visibility to managers on the status of work and progress (to do lists), as well as problem management and resolution.

Summary of follow-up meeting
The requirements gathering process had been strengthened by making use of the IEEE Software Engineering Standards for Software Requirement Specifications (SRS), to establish the defined company process. In addition, requirements were identified and prioritised as mandatory, desirable, or optional. The ARM95 tool from NASA had been successfully trialled and implemented.

The software development process for the company was defined, based on the IEEE Software Engineering Standards. The company still used the waterfall model, but was actively evaluating Rational's UML as a complement to Oracle CASE tools. The company had significantly improved their testing processes and had defined and started to use company procedures for test plans and testing. Also, formal reviews had been trialled successfully on a project with additional training planned. Configuration management had been significantly improved with the installation of Visual Source Safe. MS Word revision control was used to control documents, and a MS Access database created for a company document register. A separate testing environment has been set up to validate the software configuration before use by internal staff.

Software quality assurance was improved by the introduction of the IEEE Software Engineering Standards as the company-default process. The project breathalyser was available for use, and the draft audit guidebook due for release. Problem resolution was addressed formally, and an issues tracking database developed for use within firm #8, and available to all projects and staff.

The project management process was improved by an updated version of the project manager's handbook, incorporating details of all associated plans, based on the IEEE Software Engineering Standards. The Project Plan standard IEEE1058.1 had also been adapted as part of the company defined process for project management. Sample plans and details in the project management handbook provided basic guidance on tailoring for specific projects.

The risk management process was updated and revised, and incorporated in the project management handbook. The Practical Software Management course had been attended by staff and firm #8 intend to identify measures and indicators to help provide greater management visibility on new projects. Process establishment process was improved by the use of the IEEE Software Engineering Standards as models for the company to adapt.

All of the eight processes were re-evaluated based on observations, feedback, and limited document inspections. Basically, all of the attribute 3.1 capabilities had improved, as a direct result of the work performed since the original assessment. There were further activities to be performed, which would consolidate the gains already made. No new measurement program had been established, but the Quality Manager had implemented a more formal project post mortem process, to capture lessons learned and compare planned versus actual performance.
[Information summarised and adapted from SQI-PIP-FR08-01.doc]

Firm No.9
Summary of Strengths/Weaknesses
At firm #9, a primary strength identified was the use of automated tools which drive development within this organisation. These tools support the processes assessed and provides a basis for improving process capability in the future. Use of short, focused releases gave the team specific goals to aim for and a guide that the whole team understood and worked towards.

Source Safe and Test Track were the foundation of configuration management within the organisation and provided a solid consistent process. The schedule was used as the guide for the project to ensure that goals were reached and defined the process for development within the team.

Within projects, firm #9 allocated time for technical investigation. As well as allowing for staff to gain required skills, scheduling and product development proceeded with fewer surprises. The configuration management tool was integrated into the development environment. Also, all team members were aware of the processes used within the development of the product and were able to identify the processes and products used within the software development environment.

Performance data was being collected during the requirements elicitation process, however this data was not being used to measure performance. Requirements elicitation was not formalised but was evident in the use of storyboarding. However, this was not documented and the use of tools such as an electronic whiteboard was suggested to help with formalising the process. Defining and refining of the requirements elicitation process was conducted within the organisation. However this was very informal and not documented.

There was a standard, but undocumented process for software development. The standard process had come about through the ad hoc review of the development process.

Summary of follow-up meeting
The requirements elicitation process had been documented as part of the requirements management process. The two page standard process document followed a newly designed process definition template and was supported by a comprehensive checklist. This process was tailored and applied in the last two release cycles and feedback from customers was positive. As part of the new process, risks had been identified, tracked and monitored by assigned individuals where the risk to the next release was significant. Customers were kept informed of the status of new features and fixes by newsletter and direct contact.

Risks were identified and managed by exception, and basic measures of plan versus actual were used to monitor progress. Overall, the company placed a high priority on improving internal processes, and the effort was effective in lifting both their performance and their capability profile. The Assessment was seen as a useful mechanism to provide a focus on key areas at a time that it was critical to the company. At the time of the Follow-up meeting, firm #9 had survived a break-in at their premises, which had caused some disruption to their operations.
[Information summarised and adapted from SQI-PIP-FR09-01.doc]

Firm No.11
Summary of Strengths/Weaknesses
Firm #11's strong commitment to customer service, and focussed, tightly knit work force, enabled them to be well positioned to continue building their client base and would see them actively seeking vertical markets not currently addressed by their products. In particular the practice of requirements gathering and customer related activities were well performed, with many of these activities planned and scheduled over the course of the year using 3rd party software. While the 'front office' process of sales and marketing, and associated support services, were well performed, a coherent, end-to-end understanding of software development processes (including formalised quality assurance) was not present. These 'back office' processes were needed to form the foundation upon which the company's sales and marketing efforts were based.

Improved development capability would improve profitability by reducing time spent on development and fixing defects. At firm #11, strengths were identified in the degree of customer focus and ability to identify and track customer requirements, and the focus on developing staff competencies and building strong staff morale. Also, a coherent tool environment had been adopted to support resource allocation, work product identification and overall project management, and there existed an overall level of awareness of the business environment and of effective strategies to maintain position in the market.

However, an overall view of the product or project life cycle as a consolidating view for management was lacking, as was an effective planning of product testing, evaluation and acceptance. The absence of an effective system for assurance of quality in products or processes was also noted, and weaknesses in the system for maintaining overall integrity of the software products were recorded. There was no program for measurement of progress or risk monitoring.

Summary of follow-up meeting
Firm #11 had undergone significant internal changes in the previous 12 months that included a restructuring of senior management, and a re-statement of business goals towards new markets. Staff turnover of 70% had been recorded, with only 3 of the original 13 personnel remaining since the initial Assessment was conducted. This had significantly reduced the opportunity for implementation of initiatives that were suggested as outcomes of the Assessment. Consequently, no further actions have been undertaken by Firm #11 since the Assessment was performed, and no re-evaluation of processes was performed at the follow-up interview.
[Information summarised and adapted from SQI-PIP-FR11-01 PAT.doc]

Firm No.12
Summary of Strengths/Weaknesses
The strengths of Firm #12 lay in the professionalism of the staff and their ability to utilise innovative technologies. This firm depended on developing innovative solutions to customer problems; they derived their technologies on a detailed understanding of the application domain, and had been able to depend on their technical strengths to overcome any weaknesses in project or product management. As they become more dependent on a commercial product line rather than on time and materials contracting, these weaknesses would become more critical.

Firm #12 had the opportunity to build upon its current market strengths by developing its capability for project management, applicable both to its commercial contracting and to its product development activities. The firm's experience in operating within the methods and systems of other prime contractors demonstrated that the necessary competencies exist; it was now up to the organization to develop these capabilities internally. The organisation was aware that there was a need to introduce standard procedures across the development life cycle. The assessors noted the strengths of the domain expertise held by key personnel; the ability to define innovative solutions to customer's problems; the fact that financial risks were controlled through 'time and materials' contracting; and the control of software components through use of an effective toolset and defined procedures for configuration management. There was no effective approach to project management, and the traceability of customer requirements through the software development process was very limited. Also, in-house testing was not adequately documented and quality assurance was performed on an ad hoc and uncontrolled basis.

Configuration management was limited to source code and did not extend to other documents, and the recording of resolution of customer-reported problems was very limited. Overall, there were few assets identified that would serve to support a common ‘way of doing business’ within firm #11.

Summary of follow-up meeting
Firm #12 reported that, due to the impact of Y2K and GST, it did not have time to implement any of the recommendations from the assessment.
[Information summarised and adapted from SQI-PIP-FR12-01 PAT.doc]

Firm No.13
Summary of Strengths/Weaknesses
Firm #13’s processes that had been tailored over time to suit its business environment, but did not have high achievements of capability, making the company highly dependent on the skills and competencies of individual personnel. At firm #13 there was a strong program for building staff competence, particularly using early induction training, and through extensive use of informal mentoring; there was a focus on developing a ‘people focus’ in the staff. Recruitment of new staff posed particular problems for the company. The emphasis on managing customer expectations encouraged development of a supportive customer base.

Because of the limited process capability, the company had limited capacity to respond to significant changes in the environment, whether these were technical or commercial. Market growth, significant changes in user requirements or changes in operating systems all had the potential to place stress on the informal processes employed. Because of the generally strong corporate culture and the level of collection of useable performance data, the firm was well placed to implement successful improvements.

The processes employed by firm #13 were simple and not of high capability. However, they were well suited to the normal business operations. Firm #13 had a stable, mature product, which had good reliability in the standard configuration. This reduced the extent of rework required, limiting it to problems arising as a result of specific tailoring. Also, firm #13’s policies emphasised managing the expectations of their customers, and this had resulted in a strong market position.

There were significant changes required in the future, involving both GST and the likely adoption of a GUI, and these could impact on the current maturity. Also, there was significant potential for configuration problems to occur, particularly as customer numbers increase. There was scope for the use of an appropriate configuration management tool. The company was heavily dependent on the competency and professionalism of individual staff. There was little real visibility in the development process, and limited assurance that defined requirements were achieved. Also, firm #13 recorded and stored significant data on performance, but made very limited ongoing use of this information. At a minimum, the potential existed for better estimation of project size and cost.

Summary of follow-up meeting
The focus of process change to firm #13’s development environment was in the allocation of documentary notes to job number, indicating relevance to design and other specification issues, in the development and maintenance process. This had the

added benefit of being able to formalise a more accurate approach to the collection of certain measurement data, for management and defect prevention.

The development environment had a number of checklists in place for requirements management and risk management that had since been utilised for the collection of data to 'capture the experience' of existing practices, on a monthly basis. The checklists also provided opportunities to introduce traceability in the formal approach to testing, by using the requirements checklists in the testing of new updates, therefore providing assurance that the agreed requirements have been effectively implemented.

Higher levels of capability in both problem resolution and process establishment were achieved. In the problem resolution process, all customer reports were classified against product codes and job numbers to both co-ordinate grouping of defects and to provide a mechanism for monthly reviews and customer feedback.

The Y2K project required firm #13 to understand the effectiveness of a number of processes, and both problem resolution and process establishment improved. Data collected on job numbers and timesheets were cross-referenced and used to review accuracy of estimates of effort, and all problem resolution reports were analysed and reviewed on a monthly basis for grouping of defects within the product and to support future product design. The software product developed by firm #13 had to undergo significant revision, to satisfy legislative changes for both Y2K and GST implementations.

Comments from the feedback meeting conveyed firm #13's success in being able to initially satisfy all Y2K requirements, and be able to transfer that comprehensive understanding of the development environment and its technologies, into a more rapid resolution of GST requirements. This was reflected in the relatively short duration time of 5 months to prepare and successfully implement all known GST modifications for all current customers. Overall, this understanding allowed firm #13 to undergo rapid product changes across multiple tailored installations, whilst maintaining high levels of customer satisfaction.
[Information summarised and adapted from SQI-PIP-FR13-01 PAT.doc]

Firm No.14
Summary of Strengths/Weaknesses
Overall, firm #14 demonstrated strong process capability due to the organizational culture of the company. This was particularly reflected in software development, problem resolution and project management. There was good capability reflected in the requirements gathering process based around the strong relationships developed with the clients. Configuration management and risk management demonstrated a reasonable capability. The main area of opportunity was to formally document the existing processes. The noted strengths included long term and strong relationship with clients; strong organisational culture, fostered through staff shareholding in the company and resulting in low staff turnover; frequent and regular interaction with clients which can address concerns before they become problems; Help Desk to track problems; attention to specification, design and testing during software development; and project tracking to monitor schedule and cost slippage.

However, risks were noted: loss of key staff; growth of business leading to increase in span of control of managers and a lack of opportunity to foster the organisational culture; the current configuration management system had an exposure in change control and did not cater for documentation. Also, as quality assurance was informally performed with mental checks, firm #14 was exposed to a dependence on one person. Identification of risks associated with projects was informal and not documented, and opportunity existed to document the informal procedures and practices to allow less dependence on individuals.

Summary of follow-up meeting

Firm #14 decided to produce comprehensive implementation templates to ensure that all processes were 'canvassed' and inappropriate items deleted from an implementation. The establishment of these formal documentation procedures had commenced. Project management templates had been formalised so that there were formal documents for the three main software implementations. It was felt that due to the varied nature of the three software products that each software product should have its own template. These documents were tailored according to the software modules to be implemented.

No review of process capability was performed in the follow up process. However, it was apparent that a number of process areas had improved, for example, quality assurance with the documentation of all stated development procedures; and process establishment by the creation of standard development procedures.

Firm #14 suggested that the overall value of the Assessment was that it made them very aware of documenting all stated actions that are assumed to be performed. This would help create a standard development process, which in the past often relied on the project leaders and Managing Director to guide activities. These benefits could then be extended with the ability to tailor the development processes to suit specific needs.

While the assessment generally confirmed that firm #14's methodologies were sound, stronger documentation would stabilise the current software processes, and allow them to be supported and maintained. The add-on effect was that the firm could then see improvement opportunities arise from the enhanced knowledge acquired from understanding their own processes. Firm #14 also saw this as providing opportunities for more rapid growth in the company.

The Assessment demonstrated to firm #14 a means of linking all of the relevant processes together, to formulate a comprehensive overview of project completion—not just in product delivery. Firm #14 stated that this would ensure client satisfaction was maintained at its important level, and would ensure that projects were comprehensively completed and that client satisfaction was maintained.
[Information summarised and adapted from SQI-PIP-FR14-01 PAT.doc]

Firm No.15
Summary of Strengths/Weaknesses
The firm had successfully implemented variations of its principal product line in several organisations. The domain knowledge incorporated in the product allowed a high level of re-use, but also demanded comprehensive product knowledge to effectively market and implement the system.

Firm #15 was moving to a more market-driven environment, and this was impacting the internal product development culture that created the original product. The main product line was sold by contract, due to the extensive tailoring, data conversion and file take-up activities associated with successful implementation. The marketing company was performing the role of the de facto client. The management of customer expectations and the establishment of quantifiable quality and performance criteria were essential deliverables from the marketing organisations, and their absence was causing an adverse chain reaction in the development, implementation and support of the main product line.

Project management skills and the development of repeatable processes were two areas where the company needed to improve in order to capitalise on business opportunities. The product line approach appeared to offer a powerful engine to drive e-Commerce systems, and collaboration with a strategic business partner could provide the leverage needed to break into that market.

The firm had survived a number of major set-backs and problems. The principal product was maturing, and the development team appeared competent and motivated. The main product had many features that could address the needs of e-Commerce traders. Also, it may be necessary to protect the main software product by patent, as some of the algorithms used in the product line would need that level of protection.

Development of a generic project plan would help to standardise the planning process. A separate project plan was not documented for each project, but details were incorporated in the requirements specifications and contract. This was adequate for product development, but inadequate for providing visibility to all stakeholders in contracted projects. Proliferation of product family could be achieved by providing shrink-wrap versions of the product.

Summary of follow-up meeting
No process improvement changes were reported for any of the stated actions listed from the initial Assessment report, consequently, no re-evaluation of processes was performed. Although no process changes had been recorded in the development environment, discussion with the interviewee suggested the Assessment was very accurate in its appraisal of the company and its development processes. It was evident to some staff that some form of 'objective mentoring' was required to establish and formalise new development practices, discussed as part of the Assessment action report. It was considered appropriate to recognise some form of 'change agent' to effect the level of change and support required to be successful. The difficulty in 'unfreezing' the organisation from established development practices was recognised.
[Information summarised and adapted from SQI-PIP-FR15-01pat.doc]

Firm No.16

Summary of Strengths/Weaknesses

For the size and age of the organisation, firm #16 had already established a good level of documentation for projects and were aware of the requirement to document and define the various processes associated with software development to reduce the dependency on the Managing Director. Requirements gathering was the strongest process in the organisation probably due to the billable nature of the process and the basis on which the organisation had been founded. Risk management and project management were both performed competently. In project management there were good breakdowns and estimations of tasks and close monitoring and tracking. In risk management there was a good evaluation and documentation of project risk and ongoing monitoring and weekly evaluation. Weaknesses existed in the configuration management and quality assurance processes.

The building of a number of procedures and policies for establishment of various development processes was a definite strength. This would benefit the company long term economically, aside from articulating a quality culture. Also, a very good requirements gathering process existed that integrated risk management strategies built on a customer focus. The development environment was moving towards a conceptual framework and approach (employing Rational Methodology) that would reflect proven methods and techniques towards best practice.

Good project management practices were in place, breaking down project tasks with estimation and tracking of effort. However, a lack configuration management of all work products associated with projects exposed Firm #16 to the risk of incomplete baselines. Also, an absence of contractual signoff of requirements to ensure delivery of acceptance criteria could lead to financial problems. This needed to be taken into account for establishing better testing procedures.

The lack of an identified quality assurance strategy could have immediate impacts on Firm #16's business objectives, either through decreased customer confidence or in maintaining high quality attributes in delivered performance of the product. Problem reports needed to be tracked through to completion, to ensure that all problems are rectified. Establishment of processes, though being performed, presented an opportunity to actually plan for this establishment to ensure that it doesn’t slip and to establish it as a project. Measures of quality in terms of defects, problems, faults, etc would help to establish and monitor the quality goals and targets of the products and form part of the organisational culture. There was no structured approach to risk identification and management, and this provided an exposure to unforseen events.

Summary of follow-up meeting

Firm #16 had seen a significant increase in activity the previous past 12 months, both in projects undertaken and the release of an internally developed software product suite. This had put enormous pressure on the company to build and maintain a development architecture that was suitable and stable for the purposes of their work. This meant the company had been under constant change to improve their technologies and development ideas, to maintain their competitive advantage.

New technologies have been introduced to advance the development environment from what they referred to as their 1st generation architecture, to a 2nd generation architecture, to a current 3rd generation architecture. The focus of their efforts was in re-use of design, components and technology in E-commerce application developments.

Although this inhibited firm #16's opportunities to review previously stated Assessment actions, it was indicative of the firm's philosophy, that it will manage and improve its own development infrastructure in conjunction with its current growth. The enhanced understanding of its current software processes and inter-related business activities was reflective of the management support and commitment to improving their business, whilst setting relevant and realistic objectives for process improvement.

No review of process capability was undertaken for this follow up. However, efforts have been initiated in quality assurance and configuration management that may lead to further advancement to Level 1 in these processes in the near future, when these efforts have been fully documented. Data collection measures have been initiated in testing and code reviews to record the number of defects in generated code. Firm #16 found the Assessment to be of considerable value, in providing an objective view of their current development status, and to offer relevant improvement initiatives. The Assessment program was seen as offering an opportune way for Firm #16 to explore and discuss the success factors that contribute to process improvement, for example, being able to provide a change agent or opinion leader to initiate change, and through to the ability of the Assessment to identify opportunities for the tailoring of development processes to satisfy basic capability level ratings. [Information summarised and adapted from SQI-PIP-FR16-01 PAT.doc]

Firm No.17
Summary of Strengths/Weaknesses
Firm #17 concentrated its development on a single product. The product had a high reputation, and the firm had developed good relationships with all of its client groups. The firm followed a reasonable planning process to establish the scope of major releases of the product, though activities to achieve the plans were tracked only informally. Sound configuration management practices were in place to ensure the integrity of the product. A stable environment helped to control the risks associated with processes that did not demonstrate adequate capability.

Product development was weakened by the lack of any structured approach to system testing. In addition, the process for tracking customer-reported problems was informal and not controlled effectively. Tracking of activities was limited, and no records of the effort, costs or duration of tasks were kept, so that estimating for new releases lacked credibility. No effective processes existed for quality assurance or risk management, and while some assets to support process performance existed, there was no mechanism to identify or develop additional assets.

Summary of follow-up meeting
GST had a major impact on Firm #17's clients. To cope with the huge increase in product sales and subsequent training and support, the total number of staff had increased by 70 percent. The chief programmer had resigned (and had not yet been replaced) and a full-time tester had been appointed. As a result of the assessment, Firm #17 had commenced a SPI project to document and formalise the software development processes. To address recognised risks regarding testing, a tester had

been appointed, test plans were formulated and test logs and incidents recorded. Furthermore, Firm #17's workflow management system had been extended to integrate client registration, automated problem tracking, help desk and despatches. This system was being integrated with the development systems. This project and others were being tracked with Microsoft Outlook at the task level.

As Firm #17 had just commenced its SPI project, the capability of the target processes was not formally re-evaluated. Details relating to size of released product were being collected. Also, tasks associated with six projects were being tracked in Outlook. Firm #17 considered the assessment to be of great benefit. The proposals for action in the assessment report provided the impetus to develop a SPI program by enabling the firm to focus on a set of tasks. As well as providing a practical approach, the involvement of the 3rd party assessors provided a measure of accountability: staff were motivated to get the SPI program underway prior to the follow-up meeting.

Firm #17’s SPI program was not as advanced as it hoped, however, the improvements in the testing procedures have resulted in Firm #17 being more confident in releasing products. Also, there was more confidence to expand the development effort. The SPI program had already shown value by reducing the disruption resulting from staff turnover. On the whole, Firm #17 found the assessment provided value in motivating improvement actions. Due to phenomenal sales and support activity (due to GST introduction) and the loss of key staff, Firm #17 was not very advanced with the improvement actions taken since the assessment. However, Firm #17 was convinced the actions taken have already resulted in improvements in its product and processes.
[Information summarised and adapted from SQI-PIP-FR17-01.doc]

Firm No.18
Summary of Strengths/Weaknesses
The assessment revealed that Firm #18 had a remarkably mature process for a small business. The principal business of the organization focused around a well-defined process, based upon the firm’s methodology and quality manual. There was excellent control of initial project requirements, and changes over the course of a project were well handled, though on an individual project basis. Firm #18 effectively addressed financial risks, through undertaking work on a ‘time and materials’ basis. Project management was limited in scope but effective.

As a result of relatively rapid growth in recent years, Firm #18 faced problems in ensuring consistent application of its defined process across the life cycle. Many of its approaches to project management, while appropriate to its current environment, were limited in their use in less well-controlled environments. There was a need for a thorough review of the quality management system, to ensure that it retained its usefulness in a changing business environment. Firm #18 also needed to take more advantage of its strengths by developing effective measures for monitoring performance in terms of both productivity and product quality.

Summary of follow-up meeting
The firm’s methodology had been reviewed subsequently updated. In particular, modeling had been extended to include Object Oriented and Unified Modeling

Language concepts. All templates were updated to reflect changes. To ensure all staff became familiar with the changes, a workshop was developed and delivered. Procedures for the use of a configuration management tool were updated and dispersed through mentoring. A staff member had been given the duties of code librarian. An Enterprise-wide change request system had been designed and was in the process of being developed. Also, a software package had been introduced to help track and manage bugs and issues.

A risk assessment and management procedure had been developed. This project had a major impact on the quality management system and necessitated changes to procedures including testing, contract review and planning, and requirements control. A process for developing new procedures had been defined and a template had been developed and included in the quality manual to be used for all new procedures.

The changes implemented by Firm #18 impacted on the capability of four of the target processes: software development; configuration management; risk management; and process establishment. Many of the changes were too new to have impacted at the time of the follow-up meeting. However, the configuration management tool and error-tracking software had made it easier to manage multiple developer projects, and testing had been enhanced in terms of efficiency and quality.

Firm #18 considered that the assessment provided valuable motivation to review and improve the software development process. The assessment provided the impetus to make available resources to address the action items from the assessment report. Staff at Firm #18 also considered the assessment results provide evidence of their software process capability and therefore provide competitive advantage in formal tenders. Finally, the strengths highlighted in the assessment report improved the morale of the team by providing positive feedback about the value of process improvement. Firm #18 was convinced the improvement actions resulting from the assessment would return great value in the future by ensuring it was better placed to bid for large projects.
[Information summarised and adapted from SQI-PIP-FR18-04.doc]

Firm No.19
Summary of Strengths/Weaknesses
At firm #19, requirements gathering and analysis was performed well by experienced qualified engineers. The software development, problem resolution and risk management processes were sound. Also, software configuration management was well done. The major risks for the company related to the business development and the winning of new development orders or contracts. Another risk identified was the reluctance of clients to invest in quality assurance. Also, it was noted that software development in web time required evolutionary development with high visibility to and participation by the client. Significant investment was required to update technology to stay current or ahead of the competition. Finally, it was recognised that software development with significant re-use was a key to meeting market demands for faster availability and lower cost.

Summary of follow-up meeting

Summary of events/improvements since initial assessment

Firm #19 identified business development as a critical issue, and in the absence of a business developer resource, the company had reduced its full time staff to one, and was actively pursuing development projects. At the time of the planned follow-up meeting, firm #19 was repairing damage to servers, and external communication links resulting from a break-in at the premises. No changes to the target processes were identified at the meeting. PSP based measures were collected as part of the contract work. No repository, or formal measurement program, was in operation.

The Assessment identified the need for active business development, and a Statement of Capabilities would have been a major advantage. With contract work, the sponsor was heavily committed to revenue generating activities, but had started to document the Development Practices Guide required for the company.

Firm #19 decided to include factory testing as a milestone/deliverable in future contracts and also to use configuration audits as part of the regular internal maintenance procedures. Plans were also in place to introduce a formal system for Bug Tracking, and managing individual ‘to-do’ lists.

[Information summarised and adapted from SQI-PIP-FR19-01.doc]

Firm No.21

Summary of Strengths/Weaknesses

Requirement gathering was performed by experienced, competent senior people. Software developers were experienced, and the Software Manager was able to take a ‘hands-on’ role in monitoring and coaching the development team. Visibility within the company was adequate.

Configuration management was performed with tool support, and was based on separate guidelines for development and maintenance. The software manager was currently in a position to monitor the system. Problem resolution was managed with tool support. The software manager was close enough to the problems to enable identification of trends and repeat problems.

The major risk identified in the assessment was the dependence upon the software manager to provide the necessary oversight of the performance. This situation could provide a risk to the company if holidays or illness resulted in a lengthy absence. The weekly planning and scheduling approach was adequate for the small team now, but would cause problems for a larger team with a longer schedule. There were a number of current projects that have resources relocated according to priorities that may change from time to time. Internal R&D projects were usually the ones that had their resources moved.

The reliance on a few key personnel meant that the technical aspects of a project may exceed most of the team’s capabilities. It could also lead to overwork by key personnel, with resultant low morale, exhaustion, burnout, attrition and delays in decision-making and reviews.

Summary of follow-up meeting
The staff work long hours already, on the revenue generating tasks, and without additional resources, SPI progress was limited. Based on the examples shown to the follow-up assessor, the configuration management ratings were re-assessed at level 2. Since the assessment, limited measures of plan versus actual were recorded.
[Information summarised and adapted from SQI-PIP-FR21-01.doc]

Firm No.22
Summary of Strengths/Weaknesses
Firm #22 had a mature and disciplined approach to software development. Processes employed were for the main part well documented, and their practice was institutionalised. There was an excellent understanding in qualitative terms of the mechanisms and tools employed, and projects were well planned and effectively managed. A comprehensive quality assurance system was in place, tailored to the specific business needs of the company.

There were some specific weaknesses in the recording, tracking and resolution of problems, which was not performed in a consistent way across all projects. Measures collected for monitoring or performance evaluation were limited, and little detailed analysis of data was performed. This limited the firm's ability to encapsulate its experiences, and profit from them. The identification and prioritisation of risks was informal, though identified risks were effectively managed. The basis for the ongoing refinement and expansion of the set of process assets tended also to be informal.

The company had a strong, documented and well institutionalised process for identifying and developing business opportunities, leading to a thorough understanding of customer requirements for any individual project. Project management was effectively implemented, and was based upon documented processes and a well understood organisational culture. Good practice in software development was followed, with a focus on the architecture of the system. Designs were documented, and comprehensive system test plans developed. Software quality assurance was effective, and tailored to deliver effective outcomes based upon identified business needs. There was a reasonable process for developing, implementing and maintaining required process assets, and resources were provided for this purpose.

However, while a system for problem tracking and management existed, it was not used consistently across all projects. While identified risks were addressed in the planning process, there was no common approach to the identification and prioritisation of risks in new projects. Also, tracking and monitoring was limited to measures of effort, cost and schedule. Estimation was based primarily on moderated expert opinion, with very limited use made of historical data. Finally, opportunities for the company to learn from experience were limited to some extent, though the post project reviews helped to capture some useful information.

Summary of follow-up meeting
Communication from members of Firm #22 suggested that no follow up actions have resulted from the Assessment program. No re-evaluation of processes was performed in this follow up interview. Furthermore, comments from Firm #22 personnel

contacted for the follow-up report, suggested that no 'real benefits' had been gained from the Assessment performed, as the communication received from the RA discussion and report was limited. It was believed that the one-day assessment had highlighted minimal improvement initiatives, and was too brief to be of any value.
[Information summarised and adapted from SQI-PIP-FR22-01pat.doc]

Firm No.23
Summary of Strengths/Weaknesses
Software development in Firm #23 was generally performed so as to achieve the purpose of the processes employed. There was however considerable inconsistency across the organization in process implementation. This problem was accentuated by the distributed nature of Firm #23's organization, with development activities spread across several locations in different regions. This problem had been addressed by emphasising the professionalism and competency of staff, and there had been significant investment in staff development.

Most of the challenges faced by Firm #23 derived from the distributed nature of the organization. With project tasks being performed in multiple locations, project management was more difficult, particularly for monitoring and recording progress. Configuration management posed particular problems, while difficulties were found in quality assurance and problem resolution. The development of a consistent approach to process performance across the organization would help to address many of these issues.

Summary of follow-up meeting
An internet-based document control system had been set up but was not well supported within the firm. The level of Internet access varied considerably between the different firm locations, and this had been a major factor hindering implementation. A more formal system for approval of projects had been established, involving approval by the relevant Business Unit, with overall coordination and monitoring through a new control unit. A workflow management system was being developed but it had not yet been implemented at the time of the follow-up meeting. Difficulties had been encountered in the development and deployment of an effective problem management system. The distribution of functions across the different sites of the firm was partly responsible for these difficulties; problems were often reported in terms that were not easily understood by the group responsible for addressing them.

The establishment of a control unit had resulted in clarification of responsibilities for risk management within Firm #23. Risk management was still seen mainly as the responsibility of top-level management, and the process for managing risk remained informal. The additional control steps introduced through the establishment of the control unit and the revised project approval process addressed some of the weaknesses in the project management process.

The development of a common approach to systems development remained the principal focus of attempts to improve overall effectiveness. Until more progress was made towards the more effective integration of the whole enterprise, simple process improvement efforts may have limited success. Nonetheless, useful progress had

been made towards addressing some of the identified risk areas, and further actions were planned.
[Information summarised and adapted from SQI-PIP-FR23-01.doc]

Firm No.24
Summary of Strengths/Weaknesses
Because of its role in developing integrated, embedded software as part of a defined product, the company takes a systems engineering perspective rather than seeing itself as simply a software developer. Firm #24's sister company had ISO9001 accreditation which allowed it to operate in a culture which was quality orientated. Because much of the activity of requirements gathering and testing were located in the sister company, the firm could focus on a limited portion of the development lifecycle concerning design, implementation and unit test. The unit apparently had the strong support of the marketing group in resisting schedule pressure, thereby allowing it to concentrate on the quality of the products.

Some opportunities for changes were identified. A defined, formalised risk assessment and management process was a high priority and would allow early consideration of strategies and development of contingency plans. The capability to track project progress against defined milestones was poorly developed but simple changes in the way that the existing software tools were used would provide a cheap and effective increase in management oversight and could easily provide regular, standardised reports. There was a reported desire to change the typical project cycle time but it was apparent that the dynamics of the current four month regime was not fully understood. Again the use of the existing tools could provide data to produce a model of the current processes and a foundation for studying proposed changes. Clearly defined, the process for team leaders to follow would allow expansion and rapid training for new team leaders. Similarly, a policy of development staff acquisition that allowed some overlap in skills would afford some protection against unplanned absences and enable short-term transfers between teams.

Summary of follow-up meeting
There was no follow-up meeting as Firm #24 did not continue with their SEA membership.
[Information summarised and adapted from SQI-PIP-RA24-AR-10.doc]

Firm No.25
Summary of Strengths/Weaknesses
Firm #25 had an effective process for requirements gathering, due to the expertise and domain knowledge of the managing director. The process establishment process was also done well, with process documentation and standards in place for work products, but this was generally only applied to the software development processes and the assessors considered it should be applied across other processes.

Whilst software development was very strong, it was noted that implementation of proper testing procedures and traceability procedures, together with actual implementation of the defined process would improve this process. Project management could be improved with adequate tracking. Configuration management could be improved with proper planning. Problem resolution relied heavily on an organisational developed tool, and by managing this process its capability would improve.

The two weak areas were quality assurance and risk management. There was a basis and a culture for quality assurance within firm #25 with informal reviews and checklists in place. Implementation of the checklists and proper recording of the QA activities would improve this process. Risk management could be improved by formalising, planning, documenting and monitoring the risks involved with projects.

Without proper tracking of actual effort on project tasks, there was no early warning of any potential schedule slippage or cost overrun. There was a lack of planning and definition in configuration management activities, and the informal risk management exposed firm #25 to risks in relation to the offered warranty. Testing procedures were not formalised, and there was little traceability between specifications, design and implementation. The quality assurance checklists needed to be used within a large project in order to reduce the risk of implementation problems. This also enabled Firm #25 to evaluate the usefulness of the checklists. There was an opportunity to define and document additional processes apart from software development.

Summary of follow-up meeting
The follow-up meeting revealed that some of the assessment recommendations had been implemented, but a formal reassessment was not carried out. An initial proposal for action put forward in the Assessment was to track actual task effort for all project activities to enable an early warning system for cost and scheduling variations. This was implemented by allocating specific work task category codes to individual timesheets, which could also be tracked in parallel, through the use of actual project task monitoring using MS Project. New quality assurance checklists were established in the testing phase. No re-evaluation of process capability was performed at the follow up stage. However, several new processes with the potential for collecting measures have been implemented in the development areas at Firm #25. This included the tracking of estimated against actual task efforts and the number of defects recorded in testing. Firm #25 acknowledged the significance of formalising the testing process before any release to the clients.

Feedback from Firm #25 suggested that the Assessment provided very beneficial assistance to their development environment. Aside from providing confirmation of their current development processes, the assessment team conveyed potential benefits in the tailoring of improvement initiatives towards the strengths and weaknesses of the organisation. An example of this was to encourage the adoption of their quality assurance review checklists in testing across all projects, which not only helped verify the underlying QA process but also formalised the establishment of the testing procedures and offered an opportunity to collect test measures. It was also evident that Firm #25 had initiated change to their development processes by setting relevant and realistic objectives that could be achieved and would contribute to the future success of the organisation. It was clear from the discussion that management support for change to occur was pivotal in the degree of success that was obtained and that the improvement initiative was seen as a project itself with effective planning and control measures in place.
[Information summarised and adapted from SQI-PIP-FR25-01 PAT.doc]

Appendix I. Statistical tests relating to PIP field experiments

Table I.1 Schedule of assessments, assessors and follow-up meetings

Experiment #	Assessment date	Lead assessor	Support assessor	Follow-up date	Follow-up assessor
1	Aug 1999	A1	A3	Nov 2000	A3
2	Aug 1999	A1	A2	Not done	A5
3	Aug 1999	A3	A1	Dec 2000	A5
4	Aug 1999	A3	A2	Dec 2000	A3
5	Sep 1999	A1	A5	Apr 2000	A1
6 Not in this study – Showcase participant					
7	Sep 1999	A3	A9	Sep 2000	A3
8	Sep 1999	A3	A6	July 2000	A3
9	Sep 1999	A3	A7	July 2000	A3
10 Not in this study – Showcase participant					
11	Oct 1999	A1	A8	Nov 2000	A5
12	Oct 1999	A1	A9	Dec 2000	A5
13	Nov 1999	A1	A8	Oct 2000	A5
14	Sep 1999	A2	A6	Nov 2000	A5
15	Nov 1999	A6	A3	Dec 2000	A5
16	Nov 1999	A2	A5	Dec 2000	A5
17	Nov 1999	A1	A4	Aug 2000	A4
18	Oct 1999	A1	A4	July 2000	A4
19	Nov 1999	A3	A8	Oct 2000	A3
20 Outside scope and funding of SEAQ PIP					
21	Nov 1999	A3	A5	Nov 2000	A3
22	Nov 1999	A1	A5	Dec 2000	A5
23	Dec 1999	A1	A2	Nov 2000	A1
24	Nov 1999	A6	A4	Not done	-----
25	Nov 1999	A2	A8	Nov 2000	A5

Table I.2 List of organisational characteristics

	Year	Number of staff – employment status			Number of staff – formal education			Number of staff – role		Number of staff – experience		ISO 9001
Org #	Founded	Full time	Part time	Contract	PG	Grad	Other	Technical	Support/ admin	< 5 years	>5 years	
1	1990	7	2	0	1	1	7	3	6	2	7	No
2	1996	2	2	2	0	3	3	4	2	5	1	No
3	1990	14	2	2	4	5	9	7	11	16	2	No
4	1998	1	2	0	1	2	0	2	1	3	0	No
5	1997	4	3	1	0	7	1	4	4	6	2	No
7	1992	4	0	5	0	6	3	8	1	8	1	No
8	1990	40	30	0	50	20	0	60	10	56	14	Yes
9	1997	5	0	1	0	5	0	4	1	3	2	No
11	1996	7	0	0	3	2	2	4	3	3	4	No
12	1993	4	0	4	0	8	0	5	2	3	5	No
13	1985	12	0	1	0	9	0	10	2	7	5	No
14	1986	9	0	17	0	2	0	21	5	20	1	No
15	1994	4	4	3	1	3	0	7	4	3	8	No
16	.	5	1	0	0	5	1	4	1	1	5	No
17	1984	10	0	0	0	4	0	4	6	6	4	No
18	1992	10	0	1	6	5	0	10	1	8	3	No
19	1994	3	1	0	0	3	0	3	1	4	0	No
21	.	14	0	2	3	4	9	4	4	2	2	No
22	1987	60	2	8	70	0	0	56	14	.	.	Yes
23	1994	60	1	0	15	42	4	56	5	35	26	No
24	.	17	0	0	4	12	1	17	0	11	6	No
25	.	2	2	1	1	4	0	4	1	3	2	No

Table I.3 Target business sector frequencies

Industry Sector	Number of responses
Finance (excluding banking)	5
Insurance	2
Banking	2
Petroleum	3
Automotive	2
Public Utilities (Gas Water Electricity)	8
Aerospace	1
Telecommunications	8
Public administration	5
Consumer Goods	2
Retail	4
Distribution/Logistics	5
Defence	4
Information Technology/software	9
Health and Pharmaceutical	2
Leisure and Tourism	6
Manufacturing	6
Construction	5
Travel	1
Media (TV radio)	1
Education	2
*Mining	2
*Agriculture	1
Total	88
Note * denotes new sectors added to list by researcher during analysis of data	

Table I.4 Other target business sector responses recoded to listed sectors

Target business sector recorded by sponsor	Recoding by researcher
Security Systems	Defence
Knowledge management	IT/software
Wholesale, transport, general business	Sponsor had also selected distribution/logistics
Fleet management, bespoke software	Sponsor had also selected distribution/logistics
Internet - industry services	IT/software
Agriculture	Agriculture (new sector added to list of sectors)
Mining (2 responses)	Mining (new sector added to list of sectors)

Table I.5 List of project characteristics for each firm – actual responses

Org #	Projects in progress	Staff/project	Project duration	Cost overrun
1	3	2	6 mth/release	20.00%
2	3	2	8weeks	0%
3	1	5	2 years	
4	2	1full- time,2 part-time	18 mths	
5	4	4	90 days	20.00%
7	2	5	6mths	0
8	5-10	5-10	6-24 mths	10.00%
9	1	5	5 years so far	
11	4	3	3-6 mths	40.00%
12	3	2	2-3 mths	
13	20-50	<1	.5-15 man days	-10 to 20%
14	10	1-3	2-4 mths	1-10%
15	5	2	30 days +	100%
16	7	2	3 mths	
17	2	2-3	6 mths	Development not required to submit or work to a budget
18	5-6	2-3	12-18 mths	
19	2-3	5	6-12 mths	50
21	4	1-2	1-12 mths	?
22				
23	5	10	1-3 yrs	0
24	3	2	16 weeks	40%
25	5	1-2	2 wks-12 ,mths	~10%

Table I.6 Firm headcount, process capability level and attribute achievement for each process at assessment

Firm		Process capability levels								Process attribute achievement							
Id#	Staff FTE	RE	SD	CM	QA	PR	PM	RM	PE	RE	SD	CM	QA	PR	PM	RM	PE
1	8	2	1	1	0	1	0	1	0	7	4	4	2	3	3	6	1
2	4	1	1	0	0	1	1	1	0	6	7	1	1	3	4	3	0
3	16	2	1	1	1	1	1	0	0	9	8	8	4	10	4	4	6
4	2	2	2	3	2	3	2	1	1	11	13	14	13	14	12	9	7
5	6	1	1	1	0	1	0	0	0	5	6	2	0	2	1	0	0
7	6.5	2	2	2	1	1	3	2	2	10	10	12	10	10	14	11	10
8	55	2	2	1	1	1	2	3	1	11	10	10	11	12	12	13	11
9	5.5	1	2	3	0	3	2	0	0	8	11	14	1	13	12	1	3
11	7	2	1	1	0	0	1	1	0	12	6	3	0	1	6	4	2
12	6	0	1	1	0	0	0	0	0	2	5	2	0	0	1	1	1
13	12.5	1	0	1	0	0	1	0	0	4	5	5	3	1	5	0	1
14	17.5	1	2	1	0	2	2	1	0	7	10	6	1	8	10	4	1
15	7.5	1	1	1	0	1	1	1	0	5	5	4	1	6	4	2	1
16	5.5	2	1	0	0	1	1	1	1	9	8	2	1	8	6	5	5
17	10	1	1	1	0	0	0	0	0	5	6	2	0	1	1	0	0
18	10.5	3	2	1	2	1	2	0	1	14	12	8	11	6	13	3	6
19	3.5	1	1	2	0	1	1	1	0	6	8	10	2	7	6	6	5
21	15	1	1	1	0	1	1	0	0	7	7	5	0	7	2	2	1
22	65	3	2	2	3	1	2	1	1	13	11	9	14	6	11	7	9
23	60.5	1	1	0	1	1	1	0	0	6	7	1	4	2	5	1	1
24	17	2	2	2	1	1	1	0	0	11	8	10	8	7	2	0	0
25	3.5	2	1	1	1	0	1	0	2	8	10	3	0	2	3	1	9

Table I.7 Tests of normality for capability levels at assessment and follow-up meeting, and attribute achievement at assessment and follow-up meeting

Variable	Process	Kolmogorov-Smirnov(a)			Shapiro-Wilks		
		Statistic	df	*p*	Statistic	df	*p*
Capability level at assessment	RE	.270	22	.000	.846	22	.003
	SD	.349	22	.000	.732	22	.000
	CM	.337	22	.000	.821	22	.001
	QA	.346	22	.000	.720	22	.000
	PR	.364	22	.000	.745	22	.000
	PM	.272	22	.000	.862	22	.006
	RM	.290	22	.000	.740	22	.000
	PE	.412	22	.000	.647	22	.000
Capability level at follow-up meeting	RE	.244	22	.001	.861	22	.005
	SD	.351	22	.000	.789	22	.000
	CM	.257	22	.001	.877	22	.011
	QA	.299	22	.000	.790	22	.000
	PR	.379	22	.000	.748	22	.000
	PM	.321	22	.000	.836	22	.002
	RM	.253	22	.001	.795	22	.000
	PE	.364	22	.000	.699	22	.000
Attribute achievement at assessment	RE	.126	22	.200(*)	.975	22	.814
	SD	.145	22	.200(*)	.954	22	.377
	CM	.152	22	.200(*)	.909	22	.045
	QA	.249	22	.001	.782	22	.000
	PR	.162	22	.138	.934	22	.148
	PM	.203	22	.019	.884	22	.014
	RM	.157	22	.168	.882	22	.013
	PE	.264	22	.000	.845	22	.003
Attribute achievement at follow-up meeting	RE	.137	22	.200(*)	.967	22	.640
	SD	.180	22	.062	.911	22	.050
	CM	.169	22	.104	.918	22	.070
	QA	.243	22	.002	.825	22	.001
	PR	.167	22	.110	.927	22	.108
	PM	.219	22	.008	.875	22	.010
	RM	.156	22	.178	.910	22	.047
	PE	.250	22	.001	.851	22	.004

a Lilliefors Significance Correction
* indicates distribution is normal (p>.05)

Table I.8 Frequency of capability levels by process at initial assessment

Process	Capability level			
	0	1	2	3
Requirements elicitation	1	10	9	2
Software development	1	13	8	0
Configuration management	3	13	4	2
Quality assurance	13	6	2	1
Problem resolution	5	14	1	2
Project management	4	11	6	1
Risk management	11	9	1	1
Process establishment	15	5	2	0
Total	53	81	33	9
Percent	30.11%	46.02%	18.75%	5.11%
Levels: 0 incomplete, 1 performed, 2 managed, 3 established				

Table I.9 Comparison of process capability levels and attribute achievement
Friedman K related samples

Measure	Process	Mean rank
Capability level at assessment	RE	6.27
	SD	5.64
	CM	5.23
	QA	3.16
	PR	4.48
	PM	5.11
	RM	3.48
	PE	2.64
Friedman test statistics for capability levels	N	22
	χ^2	54.663
	df	7
	p	.000
Attribute achievement	RE	6.18
	SD	6.32
	CM	5.05
	QA	2.80
	PR	4.73
	PM	5.11
	RM	3.20
	PE	2.61
Friedman test statistics attribute achievement	N	22
	χ^2	57.692
	df	7
	p	.000

Table I.10 Wilcoxon signed ranks test of process capability levels

Process pair	*Z*	*p* (2-tailed)	Group
SD – RE	-1.508	0.132	a
CM – RE	-1.431	0.152	a
PM – RE	-2.000	0.046	*
PR – RE	-2.125	0.034	*
RM – RE	-3.337	0.001	*
QA – RE	-4.001	0.000	*
PE – RE	-3.987	0.000	*
CM – SD	-0.632	0.527	b
PM – SD	-1.134	0.257	b
PR – SD	-2.111	0.035	*
RM – SD	-3.095	0.002	*
QA – SD	-3.398	0.001	*
PE – SD	-3.750	0.000	*
PM – CM	-0.258	0.796	c
PR – CM	-1.387	0.166	c
RM – CM	-2.231	0.026	*
QA – CM	-2.707	0.007	*
PE – CM	-3.124	0.002	*
PM – PR	-1.069	0.285	c
RM – PM	-2.676	0.007	*
PM – QA	-2.804	0.005	*
PE – PM	-3.532	0.000	*
RM – PR	-1.524	0.128	d
PR – QA	-1.696	0.090	d
PE – PR	-2.295	0.022	*
RM – QA	-0.166	0.868	d
PE – RM	-1.249	0.212	e
PE – QA	-1.155	0.248	e

* significant difference at p=0.05
Note: pairs with the same group letter are not significantly different

Table I.11 Wilcoxon signed ranks test of process attribute achievement

Process pair	*Z*	*p* (2-tailed)	Group
SD – RE	-0.346	0.729	a
PM – SD	-2.625	0.009	*
CM – SD	-2.625	0.009	*
PR – SD	-2.691	0.007	*
RM – SD	-3.792	0.000	*
QA – SD	-3.551	0.000	*
PE – SD	-3.978	0.000	*
PM – RE	-2.273	0.023	*
CM – RE	-2.163	0.031	*
PR – RE	-2.326	0.020	*
RM – RE	-3.798	0.000	*
QA – RE	-3.724	0.000	*
PE – RE	-3.857	0.000	*
PM – CM	-0.445	0.656	b
PM – PR	-0.405	0.686	b
RM – PM	-3.127	0.002	*
PM – QA	-2.813	0.005	*
PE – PM	-3.109	0.002	*
PR – CM	-0.721	0.471	b
RM – CM	-2.539	0.011	*
QA – CM	-2.431	0.015	*
PE – CM	-2.790	0.005	*
RM – PR	-2.475	0.013	*
PR – QA	-1.933	0.053	*
PE – PR	-2.525	0.012	*
RM – QA	-0.153	0.878	d
PE – RM	-0.548	0.583	d
PE – QA	-0.363	0.716	d
* significant difference at p=0.05 Note: pairs with the same group letter are not significantly different			

Table I.12 Ranking of sector groups by capability levels and attribute achievement

Process	Sector #	Sector Group	Capability level		Attribute achievement	
			N	Mean Rank	N	Mean Rank
RE	1	Manufacturing, automotive, dist/logistics	8	23.81	8	24.69
	2	Public utilities and public administration	12	39.21	12	38.21
	3	Construction, mining, petroleum, agricult.	7	30.71	7	28.21
	4	Telecommunications, media	8	40.56	8	39.81
	5	Finance, insurance, banking	6	34.17	6	36.42
	6	Consumer goods and retail	4	29.75	4	26.63
	7	Defence, aerospace	4	46.50	4	39.38
	8	Information Technology, software	9	32.61	9	33.44
	9	Education, health, pharmaceutical	3	23.00	3	34.67
	10	Leisure and tourism, travel	6	34.17	6	36.42
SD	1	Manufacturing, automotive, dist/logistics	8	25.69	8	26.75
	2	Public utilities and public administration	12	40.75	12	41.79
	3	Construction, mining, petroleum, agricult.	7	30.50	7	33.86
	4	Telecommunications, media	8	40.75	8	37.38
	5	Finance, insurance, banking	6	35.33	6	35.50
	6	Consumer goods and retail	4	24.50	4	22.38
	7	Defence, aerospace	4	32.63	4	29.25
	8	Information Technology, software	9	35.33	9	38.83
	9	Education, health, pharmaceutical	3	24.50	3	11.67
	10	Leisure and tourism, travel	6	35.33	6	37.08
CM	1	Manufacturing, automotive, dist/logistics	8	32.50	8	30.69
	2	Public utilities and public administration	12	42.17	12	41.75
	3	Construction, mining, petroleum, agricult.	7	37.29	7	39.57
	4	Telecommunications, media	8	35.94	8	37.00
	5	Finance, insurance, banking	6	23.33	6	25.25
	6	Consumer goods and retail	4	25.63	4	16.63
	7	Defence, aerospace	4	32.50	4	39.75
	8	Information Technology, software	9	33.17	9	32.83
	9	Education, health, pharmaceutical	3	32.50	3	26.33
	10	Leisure and tourism, travel	6	32.50	6	34.50
QA	1	Manufacturing, automotive, dist/logistics	8	23.00	8	25.00
	2	Public utilities and public administration	12	42.17	12	36.54
	3	Construction, mining, petroleum, agricult.	7	32.36	7	33.36
	4	Telecommunications, media	8	40.06	8	39.19
	5	Finance, insurance, banking	6	33.92	6	33.83
	6	Consumer goods and retail	4	23.00	4	18.75
	7	Defence, aerospace	4	39.38	4	44.75
	8	Information Technology, software	9	35.06	9	37.72
	9	Education, health, pharmaceutical	3	23.00	3	26.50
	10	Leisure and tourism, travel	6	33.92	6	36.08
PR	1	Manufacturing, automotive, dist/logistics	8	29.50	8	29.25
	2	Public utilities and public administration	12	34.54	12	39.92
	3	Construction, mining, petroleum, agricult.	7	40.86	7	42.07

Process	Sector #	Sector Group	Capability level		Attribute achievement	
			N	Mean Rank	N	Mean Rank
	4	Telecommunications, media	8	30.13	8	33.50
	5	Finance, insurance, banking	6	36.67	6	29.50
	6	Consumer goods and retail	4	22.75	4	19.50
	7	Defence, aerospace	4	37.50	4	39.00
	8	Information Technology, software	9	37.39	9	35.56
	9	Education, health, pharmaceutical	3	27.67	3	22.83
	10	Leisure and tourism, travel	6	36.67	6	33.83
PM	1	Manufacturing, automotive, dist/logistics	8	29.06	8	28.81
	2	Public utilities and public administration	12	37.08	12	31.46
	3	Construction, mining, petroleum, agricult.	7	33.14	7	29.00
	4	Telecommunications, media	8	40.81	8	41.75
	5	Finance, insurance, banking	6	37.83	6	39.50
	6	Consumer goods and retail	4	22.63	4	23.38
	7	Defence, aerospace	4	29.63	4	30.63
	8	Information Technology, software	9	35.22	9	38.33
	9	Education, health, pharmaceutical	3	20.67	3	26.33
	10	Leisure and tourism, travel	6	37.83	6	42.67
RM	1	Manufacturing, automotive, dist/logistics	8	34.00	8	30.31
	2	Public utilities and public administration	12	28.96	12	32.50
	3	Construction, mining, petroleum, agricult.	7	27.71	7	29.64
	4	Telecommunications, media	8	43.06	8	42.63
	5	Finance, insurance, banking	6	35.33	6	32.33
	6	Consumer goods and retail	4	38.00	4	26.50
	7	Defence, aerospace	4	30.00	4	39.88
	8	Information Technology, software	9	31.78	9	32.78
	9	Education, health, pharmaceutical	3	46.00	3	41.83
	10	Leisure and tourism, travel	6	35.33	6	36.17
PE	1	Manufacturing, automotive, dist/logistics	8	26.00	8	26.75
	2	Public utilities and public administration	12	40.67	12	38.67
	3	Construction, mining, petroleum, agricult.	7	30.57	7	30.64
	4	Telecommunications, media	8	43.00	8	42.94
	5	Finance, insurance, banking	6	31.33	6	28.75
	6	Consumer goods and retail	4	26.00	4	18.88
	7	Defence, aerospace	4	34.00	4	39.00
	8	Information Technology, software	9	36.67	9	37.06
	9	Education, health, pharmaceutical	3	26.00	3	29.83
	10	Leisure and tourism, travel	6	31.33	6	35.83
	Total	for each process	67		67	

Table I.13 Comparison of capability levels of firms with private sector clients with firms without private sector clients
Mann-Whitney U tests

Process capability level	Public/private	N	Mean Rank	Sum of Ranks
For all processes	Total	22		
Requirements elicitation	Public 1	14	13.00	182.00
	Private 2	8	8.88	71.00
Software development	Public 1	14	13.25	185.50
	Private 2	8	8.44	67.50
Configuration management	Public 1	14	12.07	169.00
	Private 2	8	10.50	84.00
Quality assurance	Public 1	14	12.71	178.00
	Private 2	8	9.38	75.00
Problem resolution	Public 1	14	12.32	172.50
	Private 2	8	10.06	80.50
Project management	Public 1	14	12.82	179.50
	Private 2	8	9.19	73.50
Risk management	Public 1	14	13.21	185.00
	Private 2	8	8.50	68.00
Process establishment	Public 1	14	11.82	165.50
	Private 2	8	10.94	87.50

Table I.14 Comparison of ranks for attribute achievement public/private sector firms

Process attribute achievement	Public/private	N	Mean rank	Sum of ranks
For all processes	Total	22		
Requirements elicitation	Public 1	14	13.54	189.50
	Private 2	8	7.94	63.50
Software development	Public 1	14	12.00	168.00
	Private 2	8	10.63	85.00
Configuration management	Public 1	14	12.82	179.50
	Private 2	8	9.19	73.50
Quality assurance	Public 1	14	13.04	182.50
	Private 2	8	8.81	70.50
Problem resolution	Public 1	14	12.68	177.50
	Private 2	8	9.44	75.50
Project management	Public 1	14	12.32	172.50
	Private 2	8	10.06	80.50
Risk management	Public 1	14	13.75	192.50
	Private 2	8	7.56	60.50
Process establishment	Public 1	14	12.18	170.50
	Private 2	8	10.31	82.50

Table I.15 Comparison of ranks for capability levels: few or many target business sectors

Process capability level	Few/many sectors	N	Mean Rank	Sum of Ranks
For all processes	Total	22		
Requirements elicitation	Few 1	11	12.55	138.00
	Many 2	11	10.45	115.00
Software development	Few 1	11	12.14	133.50
	Many 2	11	10.86	119.50
Configuration management	Few 1	11	12.91	142.00
	Many 2	11	10.09	111.00
Quality assurance	Few 1	11	11.68	128.50
	Many 2	11	11.32	124.50
Problem resolution	Few 1	11	11.55	127.00
	Many 2	11	11.45	126.00
Project management	Few 1	11	11.36	125.00
	Many 2	11	11.64	128.00
Risk management	Few 1	11	11.55	127.00
	Many 2	11	11.45	126.00
Process establishment	Few 1	11	13.18	145.00
	Many 2	11	9.82	108.00

Few/many sectors is coded 1 for 1 or 2 sectors selected, 2 for more than 2 sectors

Table I.16 Comparison of ranks attribute achievement for few or many target business sectors

Process attribute achievement	Few/many	N	Mean Rank	Sum of Ranks
For all processes	Total	22		
Requirements elicitation	Few 1	11	11.77	129.50
	Many 2	11	11.23	123.50
Software development	Few 1	11	12.00	132.00
	Many 2	11	11.00	121.00
Configuration management	Few 1	11	12.82	141.00
	Many 2	11	10.18	112.00
Quality assurance	Few 1	11	11.86	130.50
	Many 2	11	11.14	122.50
Problem resolution	Few 1	11	12.59	138.50
	Many 2	11	10.41	114.50
Project management	Few 1	11	11.45	126.00
	Many 2	11	11.55	127.00
Risk management	Few 1	11	10.86	119.50
	Many 2	11	12.14	133.50
Process establishment	Few 1	11	11.95	131.50
	Many 2	11	11.05	121.50

Table I.17 Correlations importance of performance measures and capability levels and attribute achievement
Spearman rank correlation test results

Process		Importance of Measure					
		Budget	Schedule	Customer	Requirements	Productivity	Morale
Capability level- assessment							
RE	r_s	-.058	.110	-.141	.098	.052	.093
	p (1-tailed)	.399	.313	.266	.333	.410	.340
SD	r_s	-.100	.061	-.195	-.121	-.091	.211
	p (1-tailed)	.330	.394	.192	.296	.343	.173
CM	r_s	-.198	-.172	-.064	.134	-.123	-.088
	p (1-tailed)	.188	.222	.389	.275	.293	.348
QA	r_s	-.176	.112	-.274	-.102	-.019	.138
	p (1-tailed)	.216	.309	.108	.326	.467	.270
PR	r_s	-.245	-.102	-.282	.035	.059	.282
	p (1-tailed)	.136	.326	.102	.439	.397	.102
PM	r_s	.042	.182	-.140	-.097	.113	.315
	p (1-tailed)	.426	.208	.268	.334	.309	.077
RM	r_s	.452(*)	.001	-.233	.298	.001	-.056
	p (1-tailed)	.017	.498	.149	.089	.498	.403
PE	r_s	.183	.041	-.177	.145	-.072	.137
	p (1-tailed)	.207	.428	.216	.259	.375	.272
Attribute achievement at assessment							
RE	r_s	-.067	.109	-.218	.022	.098	.272
	p (1-tailed)	.383	.315	.165	.460	.331	.110
SD	r_s	-.069	.097	-.260	-.068	.016	.405(*)
	p (1-tailed)	.381	.333	.121	.382	.472	.031
CM	r_s	-.201	-.193	-.202	.015	-.107	-.064
	p (1-tailed)	.185	.194	.183	.474	.318	.388
QA	r_s	-.047	.109	-.417(*)	-.227	.128	-.007
	p (1-tailed)	.418	.315	.027	.155	.286	.487
PR	r_s	-.220	-.282	-.380(*)	.082	.004	.189
	p (1-tailed)	.163	.102	.041	.358	.492	.200
PM	r_s	.168	.225	-.210	-.082	.231	.312
	p (1-tailed)	.227	.157	.174	.358	.150	.079
RM	r_s	.316	-.075	-.251	.398(*)	.032	.122
	p (1-tailed)	.076	.370	.130	.033	.443	.294
PE	r_s	.107	-.132	-.172	.258	.005	.102
	p (1-tailed)	.318	.280	.222	.123	.492	.326

Note: 22 responses for each process for each perception

Table I.18 Process capability levels and attribute achievement at time of follow-up meeting

Firm	Process capability levels								Process attribute achievement							
Id#	RE	SD	CM	QA	PR	PM	RM	PE	RE	SD	CM	QA	PR	PM	RM	PE
1	2	1	1	0	1	0	1	0	7	4	4	2	3	3	6	1
2	1	1	0	0	1	1	1	0	6	7	1	1	3	4	3	
3	2	1	1	1	1	1	0	0	9	8	8	4	10	4	4	6
4	2	2	3	2	3	2	1	1	11	13	14	13	14	12	9	7
5	1	1	1	1	1	1	0	0	5	6	2	2	2	2	0	0
7	2	2	2	1	1	3	2	2	10	10	12	10	10	14	11	10
8	3	3	2	2	3	3	3	2	13	13	11	12	13	13	13	12
9	2	3	3	2	3	3	2	2	9	13	15	9	13	13	9	9
11	2	1	1	0	0	1	1	0	12	6	3	0	1	6	4	2
12	0	1	1	0	0	0	0	0	2	5	2	0	0	1	1	1
13	1	0	1	0	1	1	0	1	4	5	5	3	2	5	0	2
14	1	2	1	0	2	2	1	0	7	10	6	1	8	10	4	1
15	1	1	1	0	1	1	1	0	5	5	4	1	6	4	2	1
16	2	1	0	0	1	1	1	1	9	8	2	1	8	6	5	5
17	1	1	1	0	0	0	0	0	5	6	2	0	1	1	0	0
18	3	3	2	2	1	2	1	2	14	13	10	11	6	13	4	9
19	1	1	2	0	1	1	1	0	6	8	10	2	7	6	6	5
21	1	1	2	0	1	1	0	0	7	7	8	0	7	2	2	1
22	3	2	2	3	1	2	1	1	13	11	9	14	6	11	7	9
23	1	1	0	1	1	1	0	0	6	7	1	4	2	6	1	1
24	2	2	2	1	1	1	0	0	11	8	10	8	7	2	0	0
25	2	1	1	1	0	1	0	2	8	10	3	0	2	3	1	9

Table I.19 Correlations best practice survey adoption and process capability at assessment
Spearman's rho bivariate correlations

Process	r_s	p (1-tailed)	N
Requirements elicitation	.095	.364	16
Software development	-.052	.423	16
Configuration management	.073	.395	16
Quality assurance	.286	.141	16
Problem resolution	.050	.427	16
Project management	.107	.347	16

Table I.20 Software development process: mean adoption level from survey and process attribute achievement ratings

Firm #	Mean adoption level for BPS SD practices	Process attribute ratings				
		PA1.1	PA2.1	PA2.2	PA3.1	PA3.2
1	55.56	2	1	1	0	0
2	50.00	3	2	1	0	1
3	45.45	3	2	1	1	1
4	40.00	3	3	2	2	3
5	63.64	2	1	2	1	0
7	50.00	3	2	2	1	2
8	80.00	3	2	2	1	2
9	36.36	3	3	2	1	2
11	18.18	2	1	2	0	1
12		2	1	2	0	0
13	45.45	1	1	1	1	1
14	45.45	3	3	2	1	1
15	80.00	2	1	1	0	1
16	60.00	3	2	1	1	1
17		2	1	1	1	1
18	80.00	3	3	3	2	1
19		2	2	2	0	2
21	40.00	2	1	2	1	1
22		3	3	2	1	2
23		3	2	0	1	1
24	40.00	3	2	2	0	1
25		2	3	2	1	2

Table I.21 Software development process: correlations survey adoption with PIP attribute achievement
Spearman's rho bivariate correlations

Process attribute	r_s	p (1-tailed)	N
PA 1.1	.016	.477	16
PA 2.1	-.095	.363	16
PA 2.2	-.156	.282	16
PA 3.1	.083	.379	16
PA 3.2	-.253	.172	16

Table I.22 Configuration management process: mean adoption level from survey and process attribute achievement ratings

Firm #	Mean adoption level for CM practices (from survey)	Process attribute ratings				
		PA1.1	PA2.1	PA2.2	PA3.1	PA3.2
1	87.5	2.00	1.00	1.00	.00	.00
2	37.5	1.00	.00	.00	.00	.00
3	75.0	2.00	2.00	2.00	1.00	1.00
4	37.5	3.00	3.00	3.00	2.00	3.00
5	62.5	2.00	.00	.00	.00	.00
7	75.0	3.00	2.00	3.00	2.00	2.00
8	87.5	2.00	2.00	3.00	1.00	2.00
9	87.5	3.00	3.00	3.00	2.00	3.00
11	37.5	2.00	1.00	.00	.00	.00
12		2.00	.00	.00	.00	.00
13	87.5	2.00	1.00	.00	1.00	1.00
14	50.0	2.00	1.00	1.00	1.00	1.00
15	75.0	2.00	1.00	1.00	.00	.00
16	50.0	1.00	1.00	.00	.00	.00
17		2.00	.00	.00	.00	.00
18	100.0	2.00	2.00	1.00	1.00	2.00
19		3.00	2.00	2.00	1.00	2.00
21	12.5	3.00	1.00	.00	.00	1.00
22		3.00	2.00	2.00	1.00	1.00
23		1.00	.00	.00	.00	.00
24	37.5	3.00	2.00	2.00	1.00	2.00
25		2.00	.00	.00	1.00	.00

Table I.23 Configuration management process: correlations survey adoption with PIP attribute achievement
Spearman's rho bivariate correlations

Process attribute	r_s	*p* (1-tailed)	N
PA 1.1	-.130	.316	16
PA 2.1	.279	.147	16
PA 2.2	.324	.111	16
PA 3.1	.305	.125	16
PA 3.2	.206	.221	16

Table I.24 Project management process: mean adoption level from survey and process attribute achievement ratings

Firm #	Mean adoption level for BPS PM practices	Process attribute ratings				
		PA1.1	PA2.1	PA2.2	PA3.1	PA3.2
1	58.33	1	1	1	0	0
2	27.27	2	1	1	0	0
3	63.64	2	1	1	0	0
4	27.27	3	3	2	2	2
5	66.67	1	0	0	0	0
7	63.64	3	3	3	2	3
8	75.00	3	3	3	1	2
9	54.55	3	2	3	2	2
11	40.00	2	1	1	1	1
12	.	1	0	0	0	0
13	80.00	2	1	1	1	0
14	66.67	3	3	2	1	1
15	72.73	2	1	1	0	0
16	72.73	2	2	2	0	0
17	.	1	0	0	0	0
18	83.33	3	3	2	3	2
19	.	2	1	1	0	2
21	33.33	2	0	0	0	0
22	.	3	2	3	1	2
23	.	3	1	0	1	0
24	36.36	2	0	0	0	0
25	.	2	0	1	0	0

Table I.25 Project management process: correlations survey adoption with PIP attribute achievement
Spearman's rho bivariate correlations

Process attribute	r_s	*p* (1-tailed)	N
PA 1.1	.117	.333	16
PA 2.1	.315	.117	16
PA 2.2	.289	.139	16
PA 3.1	.188	.243	16
PA 3.2	.071	.398	16

Table I.26 Quality assurance process: mean adoption level from survey and process attribute achievement ratings

Firm #	Mean adoption level for BPS QA practices	Process attribute ratings				
		PA1.1	PA2.1	PA2.2	PA3.1	PA3.2
1	20.00	1.00	1.00	.00	.00	.00
2	.00	1.00	.00	.00	.00	.00
3	80.00	2.00	1.00	1.00	.00	.00
4	.00	3.00	3.00	2.00	2.00	3.00
5	.00	.00	.00	.00	.00	.00
7	25.00	2.00	2.00	2.00	2.00	2.00
8	25.00	2.00	3.00	3.00	1.00	2.00
9	25.00	1.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00
12	.	.00	.00	.00	.00	.00
13	25.00	1.00	.00	.00	1.00	1.00
14	.00	1.00	.00	.00	.00	.00
15	25.00	1.00	.00	.00	.00	.00
16	50.00	1.00	.00	.00	.00	.00
17	.	.00	.00	.00	.00	.00
18	100.00	3.00	2.00	2.00	2.00	2.00
19	.	1.00	1.00	.00	.00	.00
21	.00	.00	.00	.00	.00	.00
22	.	3.00	3.00	3.00	3.00	2.00
23	.	2.00	1.00	.00	1.00	.00
24	.00	3.00	2.00	1.00	1.00	1.00
25	.	.00	.00	.00	.00	.00

Table I.27 Quality assurance process: correlations survey adoption with PIP attribute achievement
Spearman's rho bivariate correlations

Process attribute	r_s	p (1-tailed)	N
PA 1.1	.372	.078	16
PA 2.1	.194	.236	16
PA 2.2	.306	.124	16
PA 3.1	.201	.228	16
PA 3.2	.169	.265	16

Appendix J. Quantitative and qualitative analysis supporting discussion

Table J.1 Survey discriminant analysis - case processing summary

Unweighted Cases			N	Percent
Valid			40	19.7
Excluded	Missing or out-of-range group codes		86	42.4
	At least one missing discriminating variable		33	16.3
	Both missing or out-of-range group codes and at least one missing discriminating variable		44	21.7
	Total		163	80.3
Total			203	100.0
Variables Failing Tolerance Test(a)				
Question	Within-groups variance	Tolerance	Minimum tolerance	
Q4.2	.000	.000	.000	
All variables passing the tolerance criteria are entered simultaneously. a Minimum tolerance level is .001.				

Note: values must be numeric, so responses were coded: 1=yes, 0=no, and null=missing values, all *not applicable* responses were coded as missing values.

Table J.2 Discriminant analysis summary of Yes responses and correlations for 20 highest and 20 lowest adopters.
Fishers exact test.

Question	Number of low adopters answered *yes*	Number of high adopters answered *yes*	χ^2	Exact *p*. (2-sided)	Corrected *p* value (2)
1.1	10	20	13.474	.000	.000
1.2	5	17	19.406	.000	.000
1.3	2	19	31.675	.000	.000
1.4	2	17	28.784	.000	.000
1.5	11	18	7.328	.020	.088(*)
1.6	5	15	12.629	.001	.044
2.2	1	19	42.153	.000	.000
2.4	8	16	11.536	.002	.088(*)
2.5a	8	19	19.078	.000	.000
2.5b	0	15	27.483	.000	.000
2.6a	1	18	41.614	.000	.000
2.6b	2	16	23.212	.000	.000
2.7	0	10	17.858	.000	.000
2.8	1	14	21.970	.000	.000
2.9	1	9	11.655	.001	.044
2.10	7	18	16.782	.000	.000
2.11	2	14	17.754	.000	.000
2.12	1	12	20.798	.000	.000
2.13	1	17	31.675	.000	.000
3.1	1	18	41.614	.000	.000
3.2	1	19	42.153	.000	.000

Question	Number of low adopters answered *yes*	Number of high adopters answered *yes*	χ^2	Exact *p*. (2-sided)	Corrected *p* value (2)
3.3	1	18	41.614	.000	.000
3.4	1	10	13.676	.000	.000
3.5	0	10	17.858	.000	.000
4.1	0	11	20.304	.000	.000
4.2	0	18	42.061	.000	.000
4.3	0	17	37.275	.000	.000
4.4	0	8	13.474	.000	.000
4.5	0	17	37.275	.000	.000
4.6	4	16	20.473	.000	.000
4.7	5	17	17.978	.000	.000
4.8	0	18	47.505	.000	.000
4.9	0	11	18.839	.000	.000
5.1	15	20	5.714(1)	.047	2.068(*)
5.2	7	16	8.286(1)	.010	.440(*)
5.3a	2	17	23.467	.000	.000
5.3b	0	10	24.631	.000	.000
5.4	4	18	24.698	.000	.000
5.5	0	13	21.109	.000	.000
5.6	3	20	29.565(1)	.000	.000
5.7	0	18	32.727(1)	.000	.000
5.8	0	20	40.000(1)	.000	.000
5.9	0	15	24.000(1)	.000	.000
5.10	0	8	13.474	.000	.000

Notes: all chi square values are for Fisher's exact test except for those denoted by (1) which are Pearson Chi Square. In most cases, the expected cell frequencies were less than five enabling the use of Fishers' exact test; in six cases, Pearson's chi-square value was reported.

(2) The p value is multiplied by 44 to account for inter-relatedness of questions (Bonferroni correction).

Table J.3 Comparison of means: non-COTS and COTS adoption by practice questions

Question	COTS	N	Mean Adoption	Std. Error	F	*p*
Q1.1	Non-COTS	114	.82	.036	1.226	.270
	COTS	87	.87	.036		
Q1.2	Non-COTS	86	.76	.047	.410	.523
	COTS	65	.80	.050		
Q1.3	Non-COTS	114	.48	.047	.667	.415
	COTS	85	.54	.054		
Q1.4	Non-COTS	99	.63	.049	.002	.967
	COTS	54	.63	.066		
Q1.5	Non-COTS	113	.77	.040	.215	.644
	COTS	84	.80	.044		
Q1.6	Non-COTS	104	.64	.047	1.412	.236
	COTS	67	.73	.055		
Q2.2	Non-COTS	112	.42	.047	.299	.585
	COTS	85	.46	.054		
Q2.4	Non-COTS	112	.72	.042	9.034	.003(**)
	COTS	85	.89	.034		
Q2.5A	Non-COTS	116	.78	.039	1.166	.282
	COTS	86	.84	.040		
Q2.5B	Non-COTS	84	.23	.046	.007	.934
	COTS	69	.23	.051		
Q2.6A	Non-COTS	115	.58	.046	2.236	.136
	COTS	83	.69	.051		
Q2.6B	Non-COTS	108	.60	.047	.119	.730
	COTS	83	.63	.053		
Q2.7	Non-COTS	113	.25	.041	.457	.500
	COTS	86	.29	.049		
Q2.8	Non-COTS	79	.32	.053	1.154	.285
	COTS	54	.41	.067		
Q2.9	Non-COTS	113	.09	.027	.852	.357
	COTS	85	.13	.037		
Q2.10	Non-COTS	116	.67	.044	12.516	.001(**)
	COTS	85	.88	.035		
Q2.11	Non-COTS	113	.55	.047	13.349	.000(***)
	COTS	86	.79	.044		
Q2.21	Non-COTS	114	.12	.031	6.208	.014(*)
	COTS	85	.26	.048		
Q2.13	Non-COTS	114	.54	.047	.030	.862
	COTS	84	.55	.055		
Q3.1	Non-COTS	115	.37	.045	9.109	.003(**)
	COTS	85	.58	.054		
Q3.2	Non-COTS	116	.52	.047	3.652	.057
	COTS	86	.65	.052		
Q3.3	Non-COTS	114	.54	.047	3.094	.080
	COTS	85	.66	.052		
Q3.4	Non-COTS	114	.14	.033	6.014	.015(*)
	COTS	86	.28	.049		
Q3.5	Non-COTS	114	.12	.031	9.115	.003(**)
	COTS	86	.29	.049		
Q4.1	Non-COTS	115	.11	.030	.102	.749

Question	COTS	N	Mean Adoption	Std. Error	F	*p*
	COTS	86	.13	.036		
Q4.2	Non-COTS	116	.41	.046	.660	.418
	COTS	86	.35	.052		
Q4.3	Non-COTS	116	.28	.042	2.960	.087
	COTS	85	.40	.053		
Q4.4	Non-COTS	115	.10	.029	.674	.413
	COTS	84	.14	.038		
Q4.5	Non-COTS	115	.30	.043	.800	.372
	COTS	85	.36	.053		
Q4.6	Non-COTS	115	.60	.046	.249	.619
	COTS	85	.56	.054		
Q4.7	Non-COTS	115	.40	.046	1.176	.280
	COTS	86	.48	.054		
Q4.8	Non-COTS	114	.39	.046	.201	.654
	COTS	82	.43	.055		
Q4.9	Non-COTS	66	.59	.061	.195	.660
	COTS	41	.63	.076		
Q5.1	Non-COTS	116	.84	.035	2.902	.090
	COTS	85	.92	.030		
Q5.2	Non-COTS	116	.51	.047	2.505	.115
	COTS	82	.62	.054		
Q5.3A	Non-COTS	115	.23	.040	.001	.971
	COTS	86	.23	.046		
Q5.3B	Non-COTS	24	.2917	.09478	.273	.604
	COTS	19	.3684	.11370		
Q5.4	Non-COTS	115	.65	.045	1.474	.226
	COTS	86	.73	.048		
Q5.5	Non-COTS	58	.53	.066	.210	.648
	COTS	36	.58	.083		
Q5.6	Non-COTS	116	.69	.043	4.217	.041(*)
	COTS	87	.82	.042		
Q5.7	Non-COTS	116	.34	.044	.127	.722
	COTS	86	.36	.052		
Q5.8	Non-COTS	116	.34	.044	1.881	.172
	COTS	84	.44	.054		
Q5.9	Non-COTS	115	.16	.034	5.616	.019(*)
	COTS	81	.30	.051		
Q5.10	Non-COTS	115	.16	.034	4.728	.031(*)
	COTS	85	.28	.049		
Note: * p<.05; ** p<.01; *** p<.001						

Table J.4 Analysis of Hofstede's scores by country and adoption of best practice

Hofstede scores (a)						Best Practice adoption (b)		
Country	Power distance	Individualism	Uncertainty avoidance	Masculinity	Long term orientation	Responses from	Number of responses	Average adoption
Australia	36	90	51	61	31	Overall Qld (c)	205	48%
Austria	11	55	70	79		Austria	16	53%
Belgium	65	75	94	54		Belgium	15	43%
Denmark	18	74	23	16		Denmark	17	55%
Finland	33	63	59	26		Finland	4	55%
France	68	71	86	43		France	18	65%
Germany FR	35	67	65	66	31	Germany	62	49%
Great Britain	35	89	35	66	25	United Kingdom	52	60%
Greece	60	35	112	57		Greece	18	57%
Ireland	28	70	35	68		Ireland	12	45%
Israel	13	54	81	47		Israel	11	46%
Italy	50	76	75	70		Italy	77	52%
Netherlands	38	80	53	14	44	Netherlands	30	49%
Norway	31	69	50	8		Norway	6	53%
Spain	57	51	86	42		Spain	34	44%
Sweden	31	71	29	5	33	Sweden	13	32%

Source: a. List of Hofstede scores for all countries: http://spectrum.troyst.edu/~vorism/hofstede.htm
b. Overall responses and average adoption level by country (source: ESI 1999); c. From this study

Table J.5 Correlations best practice adoption against Hofstede's scores

		Power distance	Individualism	Uncertainty avoidance	Masculinity
Adoption	Pearson Correlation	.172	-.001	.187	.211
	p (2-tailed)	.524	.997	.488	.433
	N	16	16	16	16

Long term orientation was not included scores for many of the countries were not available.

Table J.6 Comparison of best practice survey coverage to ISO/IEC 15504 processes

ISO/IEC 15504 Process group	ISO/IEC 15504 Base practices	Number of survey questions
Customer	CUS.1 Acquisition	2
	CUS.2 Supply	0
	CUS.3 Requirements elicitation	1
	CUS.4 Operation	0
Engineering	ENG.1 Development	9
	ENG.2 System and software maintenance	0
Support	SUP.1 Documentation	0
	SUP.2 Configuration management	8
	SUP.3 Quality assurance	5
	SUP.4 Verification	2
	SUP.5 Validation	0
	SUP.6 Joint review	0
	SUP.7 Audit	0
	SUP.8 Problem resolution	1
Management	MAN.1 Management	0
	MAN.2 Project management	12
	MAN.3 Quality management	1
	MAN.4 Risk management	0
Organisation	ORG.1 Organisational alignment	0
	ORG.2 Improvement	0
	ORG.3 Human resource management	1
	ORG.4 Infrastructure	1
	ORG.5 Measurement	0
	ORG.6 Reuse	0
TOTAL	24 Base Processes	43 questions

Note: Survey Q1.4 does not relate to any ISO/IEC 15504 process

Table J.7 Comparison of capability for small and large firms

Process	Size	N	Capability level		Attribute achievement	
			Mean Rank	Sum of Ranks	Mean Rank	Sum of Ranks
RE	1.00	19	11.00	209.00	10.89	207.00
	2.00	3	14.67	44.00	15.33	46.00
SD	1.00	19	10.95	208.00	10.95	208.00
	2.00	3	15.00	45.00	15.00	45.00
CM	1.00	19	11.71	222.50	11.45	217.50
	2.00	3	10.17	30.50	11.83	35.50
QA	1.00	19	10.42	198.00	10.32	196.00
	2.00	3	18.33	55.00	19.00	57.00
PR	1.00	19	11.34	215.50	11.37	216.00
	2.00	3	12.50	37.50	12.33	37.00
PM	1.00	19	10.84	206.00	10.82	205.50
	2.00	3	15.67	47.00	15.83	47.50
RM	1.00	19	11.00	209.00	10.82	205.50
	2.00	3	14.67	44.00	15.83	47.50
PE	1.00	19	11.00	209.00	10.71	203.50
	2.00	3	14.67	44.00	16.50	49.50
22 firms						

Note: Size=1 indicates <=50 staff FTE; size=2 indicates >50 staff FTE.

Table J.8 Qualitative analysis of factors relating to assessment

Factor/Issue	N	1	2	3	4	5	7	8	9	11	12	13	14	15	16	17	18	19	21	22	23	24	25
Managing Director attended assessment	14	Y	Y	N	Y	Y	Y	N	N	Y	Y	Y	Y	N	Y	N	Y	Y	N	N	Y	N	Y
Needed Tools for CM or to enhance /extend existing tools	7	Y	Y	Y		OK			Ok			Y	Y		Y	Y							
Need system/tool to record/track problems	8	Y			Y		Y	Y			Y				Y			Y				Y	
Testing needed to be formalized	10		Y					Y		Y	Y	Y			Y	Y		Y				Y	Y
Shortage of available staff	3			Y				Y				Y											
Rely on competent staff and informal standards (rather than documented processes)	13	Y	Y	Y	Y		Y		Y		Y	Y	Y			Y			Y		Y		Y
ISO9000 in progress (complete*)	5/4	Y	Y	Y			Y	*									Y		Y	*		Y	
Intranet development underway to enhance communication	5	Y			Y		Y												Y		Y		
Current situation OK, but need formalization as growth expected	8	Y	Y		Y	Y			Y				Y		Y		Y						
COTS developer	8	Y	Y	N	Y	Y	N		Y	N		Y	N	Y		Y							
None or insufficient measures related to problems (bugs)	8	Y	Y	Y		Ok	Y					Ok	Ok			Y	Y	Y		Y			
None or insufficient measures related to development effort	14	Y	Y	Y	Y	Y	Y		Y			Ok		Y	Ok	Y	Y	Y		Y		Y	Y
Factor/Issue		1	2	3	4	5	7	8	9	11	12	13	14	15	16	17	18	19	21	22	23	24	25

Table J.9 Advice provided by assessors at assessment

Advice	Firms	N
Standards and Guidelines		
PMBOK	#4; #8; #9; #19; #21	5
SWEBOK	#9	1
ARM-95 (NASA) verification of requirements	#2; #4; #8	3
NASA audit guidebook	#1; #8; #19	3
NASA software quality model	#9	1
Mil-Std 498 risk severity and priority classifications IEEE standards (for software requirement spec standards)	#19; #25	2
ISO 15504 part 5 templates	#7; #19	2
IEEE 1058.1 project plan	#7; #19,	2
IEEE 1016 Recommended Practice for Software Design Descriptions	#1	1
ISO 9126 Software Product Quality Evaluation	#15	1
ISO/IEC 12207 life cycle model	#15; #19;	2
ISO9000	#1; #22, #24	3
Tools/software		
MS Outlook	#17; #24	2
Visual SourceSafe (CM management)	#1; #8; #9; #12; #19; #21; #24	7
MS Project	#1; #2; #7; #9; #16; #19; #21; #24	8
Test track / team track	#1; #9/#24	3
PASS-C code analyser	#1; #4	2
Evolutionary Rapid development model/; Evolutionary life cycle approach/'see SPC document'	#1/#2/#19	3
PR Tracker	#21	1
TRIM97 – risk management tool	#24	1
RCS source control tool; /RCS & SCCS	#16/#11; #22;	3
Bug Track from Seapine, PR Track and Visual Intercept;/ Bug Track	#19/#8	2
List of available tools for CM and change control: http://www.iac.honeywell.com/Pub/Tech/CM/CMTools.html	#11; #13	2
Techniques/approaches/methodologies		
Project breathalyser – URL?	#4; #8; #9	3
Earned value	#8; #15; #16; #21; #25	5
Balanced Score Card	#8	1
Personal Software Process PSP	#8	1
Incremental model development process	#15	1
Product Line Development approach http://Interactive.sei.cmu.edu/Features/Features.htm	#24	1
X-model for development;/ 'X-model superseded by Diamond model'	#9/#19	2
PSM Practical Software Measurement	#7; #8; #9	3
Principal Best Practices http://www.spmn/com/best_practices.html	#15	1
SEPO website for measurement plan example; /for process library	#8/#19	2
QA Partner	#15	1

Table J.10 Summary of themes from final reports

Factor/Issue	N	1	2	3	4	5	7	8	9	11	12	13	14	15	16	17	18	19	21	22	23	24	25
Processes too new to be used yet	4		-	Y				Y							Y						Y	-	
Mentoring would have helped	3						Y	Y						Y									
Business problems got in the way:	7																						
Restructuring										Y													
Relocated						Y	Y																
Changed business focus		Y					Y																
Family illness					Y																		
Break in at premises									Y									Y					
Staff turnover problems	3	Y								Y						Y							
Y2K	3										Y	Y				Y							
GST	3										Y	Y				Y							
Improved testing	7			Y				Y				Y			Y	Y		Y					Y
RAPID valuable	9			Y					Y			Y	Y	Y	Y	Y	Y						Y
More measures	7					Y	Y					Y			Y	Y		Y	Y				
No action taken	1																			Y			
Difficult to implement SPI	1																				Y		

Table J.11 Extent of improvement, staff size and PIP program outcomes

Id#	Group	Staff	RE	SD	CM	QA	PR	PM	RM	PE	Outcome of PIP program
9	1	5.5	1↑	2↑	3	0↑	3	2↑	0↑	0↑	Improved 6 processes a total of 9 levels
8	1	55	2↑	2↑	1↑	1↑	1↑	2↑	3	1↑	Improved 7 processes a total of 8 levels
18	1	10.5	3	2↑	1↑	2	1	2	0↑	1↑	Improved 4 processes a total of 4 levels
5	1	6	1	1	1	0↑	1	0↑	0	0	Improved 2 processes a total of 2 levels
13	1	12.5	1	0	1	0	0↑	1	0	0↑	Improved 2 processes a total of 2 levels. GST and Y2K impact
21	1	15	1	1	1↑	0	1	1	0	0	Improved 1 process 1 level
23	2	60.5	1	1	0	1	1	1	0	0	Improved attribute. Inhibited by multiple sites
14	3	17.5	1	2	1	0	2	2	1	0	Improved QA, PE processes, and documentation
16	3	5.5	2	1	0	0	1	1	1	1	Increase in staff, # of projects.
7	4	6.5	2	2	2	1	1	3	2	2	Relocated. Improved CM
25	4	3.5	2	1	1	1	0	1	0	2	Some changes implemented
3	4	16	2	1	1	1	1	1	0	0	Adopted new methodology. Too new to assess
19	4	3.5	1	1	2	0	1	1	1	0	Disrupted by break-in at premises. Reduced operation.
1	4	8	2	1	1	0	1	0	1	0	Business focus change, sold product distribution rights
17	4	10	1	1	1	0	0	0	0	0	Lost key staff. GST big impact
4	5	2	2	2	3	2	3	2	1	1	Major non-business issue affected owner
22	5	65	3	2	2	3	1	2	1	1	1 day assessment too brief to be valuable
11	5	7	2	1	1	0	0	1	1	0	Management restructure. Changed business focus
15	5	7.5	1	1	1	0	1	1	1	0	Need mentoring, difficult to unfreeze current practices
12	5	6	0	1	1	0	0	0	0	0	Too busy due to Y2K and GST
24	6	17	2	2	2	1	1	1	0	0	SEA membership lapsed, no follow-up meeting held
2	6	4	1	1	0	0	1	1	1	0	Firm ceased to operate
Note: ↑ indicates the process capability improved											

Table J.12 Comparison of elapsed time (assessment to follow-up meeting) with type of reassessment

Reassessment Type	N	Mean	Std. Dev	Std. Error	95% Confidence Interval for Mean		Min	Max
					Lower Bound	Upper Bound		
0 Informal	11	13.0909	2.11918	.63896	11.6672	14.5146	9.00	16.00
1 Formal	9	10.7778	2.10819	.70273	9.1573	12.3983	7.00	14.00
Total	20	12.0500	2.37254	.53052	10.9396	13.1604	7.00	16.00

Table J.13 ANOVA analysis: comparison of elapsed time against type of reassessment

	Sum of Squares	df	Mean Square	F	*p*
Between Groups	26.485	1	26.485	5.925	.026
Within Groups	80.465	18	4.470		
Total	106.950	19			

Table J.14 Correlations of capability level and attribute achievement at initial assessment with extent of improvement for each process

Spearman's rho	RE	SD	CM	QA	PR	PM	RM	PE
Capability Level								
r_s	.124	.750(**)	.164	-.101	-.502	.100	-.368	.204
p (1-tailed)	.376	.010	.337	.398	.084	.399	.165	.299
N	9	9	9	9	9	9	9	9
Attribute Achievement								
r_s	.435	.714(*)	.583(*)	-.183	-.104	-.044	-.081	.570
p (1-tailed)	.121	.015	.050	.319	.395	.456	.418	.054
N	9	9	9	9	9	9	9	9

* Correlation is significant at the 0.05 level (1-tailed).
** significant at .01

Table J.15 Number of firms with and without Managing Director for each outcome group

Outcome Group	Number of firms with Managing Director present	Number of firms without Managing Director present
1 Capability level	3	2
2 Attribute achievement	1	0
3 Processes improved	2	0
4 Limited improvement	4	2
5 No improvement	3	2
6 Withdrawn from PIP	1	1

Table J.16 List of assessments, assessors and outcome group

Firm #	Assessment date	Lead Assessor	Support Assessor	Follow-up date	Follow-up Assessor	Outcome group
1	Aug 1999	A1	A3	Nov 2000	A3	4
2	Aug 1999	A1	A2	Ceased operation	A5	6
3	Aug 1999	A3	A1	Dec 2000	A5	4
4	Aug 1999	A3	A2	Dec 2000	A3	5
5	Sep 1999	A1	A5	Apr 2000	A1	1
7	Sep 1999	A3	A9	Sep 2000	A3	4
8	Sep 1999	A3	A6	July 2000	A3	1
9	Sep 1999	A3	A7	July 2000	A3	1
11	Oct 1999	A1	A8	Nov 2000	A5	5
12	Oct 1999	A1	A9	Dec 2000	A5	5
13	Nov 1999	A1	A8	Oct 2000	A5	1
14	Sep 1999	A2	A6	Nov 2000	A5	3
15	Nov 1999	A6	A3	Dec 2000	A5	5
16	Nov 1999	A2	A5	Dec 2000	A5	3
17	Nov 1999	A1	A4	Aug 2000	A4	4
18	Oct 1999	A1	A4	July 2000	A4	1
19	Nov 1999	A3	A8	Oct 2000	A3	4
21	Nov 1999	A3	A5	Nov 2000	A3	1
22	Nov 1999	A1	A5	Dec 2000	A5	5
23	Dec 1999	A1	A2	Nov 2000	A1	2
24	Nov 1999	A6	A4	SEA membership lapsed		6
25	Nov 1999	A2	A8	Nov 2000	A5	4

References for appendices

Bonate, PL 2000, *Analysis of pretest-posttest designs*, Chapman & Hall.

Coakes, SJ & Steed, LG 1996, *SPSS for Windows: Analysis without Anguish*, Jacaranda Wiley.

ESI 1995, *1995 Software Best Practice Questionnaire Analysis of Results*, ESI-PIA-960464V5, European Software Institute, Bilboa.

ISO/IEC TR 15504-2 1998, *Information technology -Software process assessment - Part 2: A reference model for processes and process capability*, ISO/IEC TR 15504-2:1998(E).

Miller, J 2004, 'Statistical significance testing - a panacea for software technology experiments?' *The Journal of Systems and Software*, vol. 73, pp. 183-92.

Poulsen, J & French, A 2002, *Discriminant Function Analysis*, San Francisco State University, viewed 14 Dec 2004, <http://userwww.sfsu.edu/~efc/classes /biol710/discrim/discrim.htm>.

Selvanathan, EA, Selvanathan, S, Keller, G & Warrack, B 2000, *Australian Business Statistics*, 2nd edn, Nelson Thomson Learning, South Melbourne.

Siegel, S 1956, *Nonparametric Statistics: for the Behavioral Sciences*, McGraw-Hill Book Company, Inc., Tokyo.

SPSS Inc. 2004, *SPSS*, 2004.