

Which version art thou? Configuration management in engineering education.

Michael D. Jokic

University of Southern Queensland, Toowoomba, Australia
michael.jokic@usq.edu.au

Andrew P. Wandel

University of Southern Queensland, Toowoomba, Australia
andrew.wandel@usq.edu.au

Alexander A. Kist

University of Southern Queensland, Toowoomba, Australia
kist@ieee.org

***Abstract:** Configuration management is a set of principles and practices used in engineering to track the history of changes made to project artefacts and products. In this paper we outline our vision for incorporating the industry practices and principles of configuration management in the University engineering environment. We discuss how revision control software tools will mitigate the “sorry, that was the wrong version” issues in undergraduate courses, support the management and quality assurance of postgraduate and staff research projects, and foster collaboration in University projects at all levels. We also see the software tools being used to help students and staff to identify the relative effort of team members in undergraduate projects for assessment purposes. An implementation program is proposed, which consists of a series of projects, leading to the establishment of a University-wide capability and beginning with a test and evaluation exercise. Our pilot project will test the open-source tools and our integration strategies via a second-year cohort consisting of on-campus and distance education students. We discuss the importance of the pilot project, our strategies for examining the expected improvement in student experiences, and the project time-frames. The implementation program will ultimately provide students with industry-relevant knowledge and skills, and staff with effective capabilities to support research projects and project-based courses.*

Introduction

Background

Configuration management is an integral part of quality assurance activities necessary for the delivery of identifiable and traceable products and designs. It is essentially a set of principles and practices (often supported by software tools) used in engineering to track the history of changes made to project artefacts and products. There are a number of standards available across the world that support the application of best practice in configuration management including MIL-STD-973 (USDoD, 1992), ISO 10007:2003 (ISO, 2003), STANAG 4159 (NATO, 1991) and IEEE Std. 828-2005 (IEEE, 2005). The principles and practices expounded by these standards are broadly adopted and modified in various engineering organisations. Indeed, as configuration management is a key element of quality assurance practices, it is also integral to organisations being certified under ISO 9001 or CMMI (Capability Maturity Model Integration).

At one end of the configuration management spectrum, versioning of artefacts might be done by periodically renaming copies or manual listing of amendments (which we consider to be typical student approaches if any versioning is done at all). At the other end of the spectrum formal, reviewed and accepted practices are defined in a configuration management plan and supported by tools. These management plans represent one of the learning materials for new graduates and important references for established staff. As the importance of configuration management is well established by its broad application in industry and there is a clear precedent for new graduates to learn these practices, it seems both logical and beneficial to integrate the practices into student projects and university activities.

Configuration management practices and tools are common to software and systems engineering environments and their importance are frequently highlighted (Hinsen, Laufer, & Thiruvathukal, 2009; Spinellis, 2005). Hinsen *et al.* (2009) discuss various aspects of commonly used configuration management tools and the available architectures for their implementation. Popular open-source software tools used to support configuration management include *CVS* (Free Software Foundation Inc., 2005) and *Subversion* (Collins-Sussman, Fitzpatrick, & Pilato, 2008) together with their associated clients.

In the educational context, Reid and Wilson (2005) investigate the application of *CVS* as a mechanism for assignment submission in computer science and conclude that it is important for such tools to be introduced early in the degree program to better prepare students to work in industry. Glassy (2006) demonstrates how *Subversion* can be used to investigate the practices and approaches adopted by students in the development of code. In our work, we also aim to exploit the potential insight provided by *Subversion* into the evolution of student project artefacts to see how the projects evolve and examine the contribution of each team member. Milentijevic *et al.* (2008) present several different models for the application of configuration management tools in project-based learning. In this case, the authors applied one of the discussed models using *CVS* to a group of 21 electronic engineering students studying a VLSI digital signal processing course. They noted that the system facilitated student cooperation and aided in the evaluation of individual contributions.

Current work

We seek to examine the potential benefits of applying configuration management tools and practices to student projects in multidisciplinary engineering courses with large student cohorts consisting of on-campus and external students. Ultimately, we expect to establish a centrally-managed university capability that can be used for team-based engineering courses and by postgraduates and staff to support their research projects. Many within the faculty are aware of the importance of configuration management and some have experience with current open-source tools. We aim to use the available knowledge and university infrastructure to make the tools widely accessible for students and staff.

Figure 1 shows the three groups within the University that are expected to benefit from establishing a centrally-managed configuration management (CM) system. It is important to note that while much of the discussion will focus on the software tools, the effective use of CM also covers practices, principles and contexts. For example, students should ultimately become aware of why CM is important in the Engineering profession. *Subversion* has been selected as the candidate open-source software tool to support our CM system. This selection has been made based on the knowledge and experience of both the authors and staff within the Division of Information and Communications Technologies at the University. Importantly, *Subversion* is a modern centralised configuration management support tool that is actively developed, widely used and well-supported outside of the university.

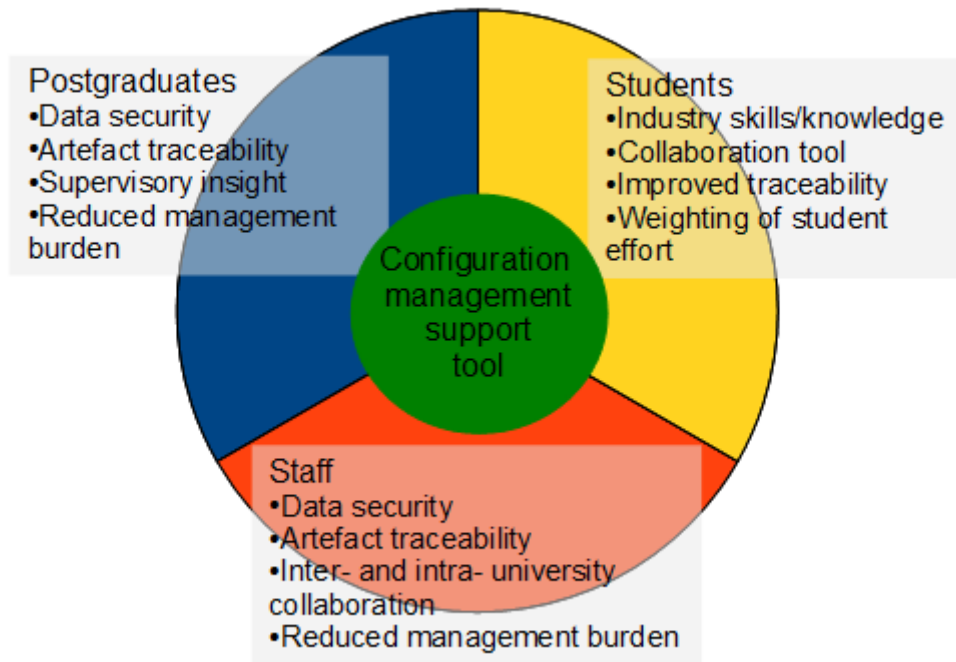


Figure 1: Summary of benefits to university community

Benefits

Figure 1 shows the hypothesised benefits we have identified for the three university populations of interest during the implementation program. Further explanation of some of the listed benefits is provided below.

- Data security** – once the first set of data is committed to the *Subversion* repository, it will not be lost assuming the server is appropriately backed-up. Any changes or modifications prior to the first commit are tracked and can be reversed at any time.
- Supervisory oversight** – people working on a particular area can all have access to the files and history of a particular project. This means that it is not necessary to send or move a file (plus any supporting files) for a supervisor to review. The supervisor can be advised of changes simply by hyperlink or by pulling down the latest copy from the repository.
- Traceability** – the entire history of electronic artifacts are neatly logged allowing for review of any changes made to key artifacts. Any version can be retrieved and compared. In the case of text files and word documents, it is possible to directly compare the differences between file versions.
- Collaboration** – the repository is a central location allowing project participants to push up and pull down updates as required. Any changes are tracked so that multiple people can potentially be working on the same file without losing track of the latest version.
- Weighting of contributions** – student changes are tracked in the system, which makes it possible to observe the number of change contributions and also to examine the depth of the contribution. *Subversion* will provide course staff and students with the ability to clearly identify any major issues with evidentiary support.
- Reduced management burden** – the proposed tools eliminate the need to manually track changes in relation to electronic artifacts. The tools will keep a record of all file versions simply via a well-tested interface, thereby eliminating the need to rename files and make unnecessary copies.
- Industry skills/knowledge** – the tools are widely applied in many engineering (particularly software-intensive) projects. This is the primary motivation for employing configuration management tools and principles within team-based projects.

Implementation

As we noted earlier, our goal is to establish a broadly used and accessible capability within the University that leverages existing resources and knowledge. Our current plan to achieve this goal involves the completion of three mini-projects, each with a well defined set of deliverables and milestones. As with the development of any capability, resourcing is a major planning and management focus. To achieve the final goal we have proposed the following mini-projects:

1. Pilot project
2. Course capability establishment project
3. Research support project

A brief summary of the projects within the program is provided in Table 1. More detail regarding the pilot project planned for execution in 2010 is presented in the next section.

Table: 1 Implementation program summary

Project	Description	Time-frame	Stakeholders
Pilot	<ul style="list-style-type: none"> •Resources – faculty and ICT staff time •Aim – to investigate the benefits to USQ course; the effectiveness of guidance and implementation; identify infrastructure limitations 	2010 – 6 months	Engineering studies course examiner Engineering studies course staff ICT infrastructure staff Project champions
Course capability establishment	<ul style="list-style-type: none"> •Resources – faculty and ICT staff time; funding for computer infrastructure and maintenance •Aim – to make the capability accessible to all faculty courses; develop suitable support materials and course content; enhance current curriculum to include CM and other QA practices. 	2011 – 6-12 months	Head of Engineering studies Engineering studies staff Dean Engineering studies course examiner ICT infrastructure staff Project champions
Research support	<ul style="list-style-type: none"> •Resources – faculty and ICT staff time; potentially funding for computer infrastructure and maintenance •Aim – to make a CM capability available to all research students and staff; prepare appropriate training materials; plan for future directions and needs. 	2011 – 6-12 months	Associate Dean of research Dean ICT infrastructure staff Project champions

Pilot project

Description

The pilot project is planned for completion at the end of semester 2, 2010. The project will be used to examine the proposed infrastructure, software tools and supporting study materials in preparation for broader application within the faculty. A second year engineering studies course in the Faculty of Engineering and Surveying at the University of Southern Queensland is the basis for the pilot project.

This course has a student population of about one hundred and fifty drawn from all engineering disciplines in the faculty, consists of on-campus and external students, is team-based and uses a problem-based learning approach. The course demonstrates the importance of numerical computation to solving engineering problems and requires students to complete software-intensive projects. In previous offerings of this course, there have been examples of students enquiring about unexpected results leading to the discovery that they submitted the wrong version of their materials. This has been a strong motivator for the pilot project.

Research objectives

We hypothesise that the following questions will be investigated during the pilot project:

- What are the limitations of the implemented CM system?
- Are improvements in the provided guidance and references needed?
- How effectively can the *Subversion* logs and student commits be used to assess student contributions to team-based projects?
- Does the CM tool improve team effectiveness and cohesion?
- Is the quality of the final product enhanced by the CM tool?
- Do the students gain an appreciation of CM, its importance and benefits?

These questions will be answered through a combination of direct observation by course and project staff, surveys, forum discussions and student focus-groups.

System overview

Figure 2 provides a high-level description of the configuration system proposed for the pilot project and (most likely) the centrally-managed capability. It is built on existing USQ IT infrastructure and open-source software. This aspect is not revealed in detail by the figure. We are working closely with our Division of Information and Communication Technology to put the computing infrastructure together for the pilot project. While the open-source software is free and well-supported, it will be necessary to establish a funding stream to maintain the capability and computing infrastructure in the long term. The pilot project will aid in securing this funding.

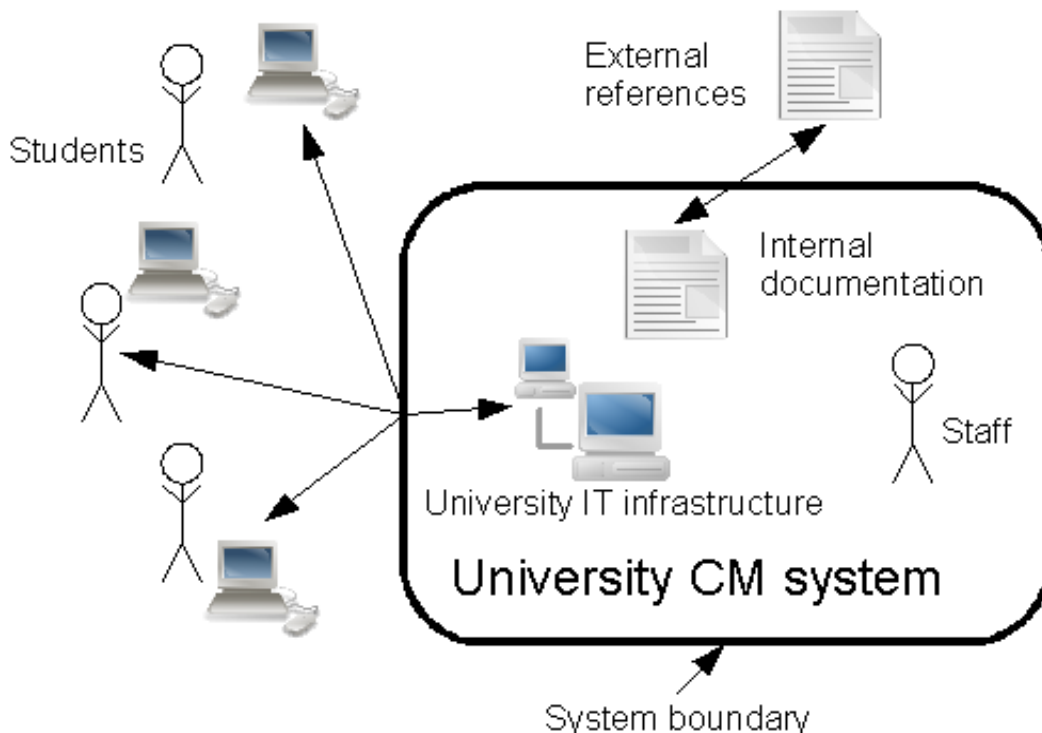


Figure 2: Pilot project system

The University CM system consists of:

- University computing infrastructure + open-source software
 - Documentation developed internally to provide background and guidance on CM and Subversion (this material will reference existing external materials available on the web, where appropriate)
 - Faculty staff that will provide guidance, support and system maintenance during the pilot project.
- The system boundary defines what elements will be managed and maintained during the pilot project by university staff.

Students will be introduced to the CM system via the course moodle site. The course moodle site will provide access to the server running *Subversion*, the references needed to use the system and discussion forums for seeking support and providing feedback. Each student will be required to download a Subversion client (preferably *TortoiseSVN* for windows users) to interface with the CM tool. Students will be expected to use the provided capability to help manage all their electronic artefacts and to use the guidance provided to effectively manage the development of their reports.

Future directions

Once a centrally-managed CM capability is established for use in courses and research projects, we will look into broadening the team-based, multidisciplinary courses to include other aspects of engineering management and quality assurance practices. This may include exploring the use of project management tools such as *Trac* (Edgewall Software, 2010) to aid with project management and product development. This particular example integrates nicely with the configuration management tool *Subversion* allowing the principles of engineering management to be incorporated into team-based projects with minimal overhead. We expect that minor changes to the engineering studies curriculum could allow students to evolve their skills and knowledge in engineering management from first-year through to final-year courses.

Conclusions

A number of important standards exist that describe 'best practice' for configuration management in engineering organisations. This combined with its broad application in industry demonstrates its importance in engineering and to engineers. We argue that the benefits afforded to engineering projects by configuration management practices and principles can translate easily to student and research projects within the University. Effectively incorporating configuration management into the USQ engineering studies stream will be undertaken via an implementation program consisting of three projects. By the end of Semester 2, 2010 we will have completed a pilot project involving one hundred and fifty students in a USQ course. The project will answer a number of important questions regarding the effective integration of configuration management into a multidisciplinary course with on-campus and external students, as well as the impact on student learning and practices.

References

- Collins-Sussman, B., Fitzpatrick, B. W., & Pilato, C. M. (2008). Version Control with Subversion Retrieved from <http://svnbook.red-bean.com/>
- Edgewall Software. (2010). trac Retrieved 15/07/2010, from <http://trac.edgewall.org/>
- Free Software Foundation Inc. (2005). Version Management with CVS Retrieved from <http://ftp.gnu.org/non-gnu/cvs/source/feature/1.12.13/cederqvist-1.12.13.pdf>
- Glassy, L. (2006). Using version control to observe student software development processes. J. Comput. Small Coll., 21(3), 99-106.
- Hinsen, K., Laufer, K., & Thiruvathukal, G. K. (2009). Essential Tools: Version Control Systems. Computing in Science & Engineering, 11(6), 84-91.
- IEEE. (2005). IEEE Standard for Software Configuration Management Plans (Vol. 828).
- ISO. (2003). Quality management systems -- Guidelines for configuration management (Vol. 10007). Geneva, Switzerland: International Organization for Standardization.
- Milentijevic, I., Ciric, V., & Vojinovic, O. (2008). Version control in project-based learning. Comput. Educ., 50(4), 1331-1338. doi: <http://dx.doi.org/10.1016/j.compedu.2006.12.010>
- NATO. (1991). NATO Material Configuration Management Policy and Procedures for Multinational Joint Projects (Vol. 4159).
- Proceedings of the 2010 AaeE Conference, Sydney, Copyright © M.D. Jokic, A.P. Wandel and A.A. Kist, 2010

- Reid, K. L., & Wilson, G. V. (2005). Learning by doing: introducing version control as a way to manage student assignments. Paper presented at the Proceedings of the 36th SIGCSE technical symposium on Computer science education, St. Louis, Missouri, USA.
- Spinellis, D. (2005). Version control systems. *Software, IEEE*, 22(5), 108-109.
- USDoD. (1992). Military Standard - Configuration Management (Vol. 973).

Acknowledgements

The authors wish to acknowledge that they have been supported in the presentation of this work by the Engineering Education Research Group at the University of Southern Queensland.

Copyright statement

Copyright © 2010 M.D. Jokic, A.P. Wandel and A.A. Kist: The authors assign to AaeE and educational non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to AaeE to publish this document in full on the World Wide Web (prime sites and mirrors) on CD-ROM or USB, and in printed form within the AaeE 2010 conference proceedings. Any other usage is prohibited without the express permission of the authors.