An Evaluation of Electronic Individual Peer Assessment in an Introductory Programming Course

Michael de Raadt, David Lai, Richard Watson

Dept. Mathematics and Computing
University of Southern Queensland
Toowoomba, Queensland, 4350, Australia

{deraadt,lai,rwatson}@usq.edu.au

Abstract

Peer learning is a powerful pedagogical practice delivering improved outcomes over conventional teacher-student interactions while offering marking relief to instructors. Peer review enables learning by requiring students to evaluate the work of others. PRAISE is an on-line peer-review system that facilitates anonymous review and delivers prompt feedback from multiple sources. This study is an evaluation of the use of PRAISE in an introductory programming course. Use of the system is examined and attitudes of novice programmers towards the use of peer review are compared to those of students from other disciplines, raising a number of interesting issues. Recommendations are made to introductory programming instructors who may be considering peer review in assignments.

Keywords: Introductory programming, assessment, peer review.

1 Introduction

Peer learning offers the opportunity for students to teach and learn from each other, providing a learning experience that is qualitatively different from usual student-teacher interactions (Saunders, 1992). Evaluation is a higher-order thinking activity (Anderson et al., 2001). Peer review encourages students to evaluate the work of others and reflect on their own work. Combined with other forms of online communication, peer review can encourage a community of learning, reducing student isolation and further encouraging higher-order thinking (Brook & Oliver, 2003). Peer review can shift instructor workload from marking to other teaching activities (de Raadt, Toleman, & Watson, 2006). Peer review used in regular assignments can increase student retention (de Raadt, Loch, & Addie, 2006).

Peer review can occur in different ways: in person or electronically, between individuals or within teams, over a period or for a single task. Much peer-review literature relates to assessing peers on contribution to work completed in groups. In online peer-review research, focus is often on online discussion, with involvement in discussion used as a means of assessment (Prins, Sluijsmans, Kirschner, & Strijbos, 2005). The system used in this study, referred to as PRAISE, creates new peer-review relationships between individuals for each assessment item. Reviews focus on student-submitted documents which are provided anonymously (double-blind) to peers for review.

This study evaluates the use of PRAISE in an introductory programming course. Student attitudes to using the system have been measured. Use of the system by novice programmers is described from system statistics. Aspects of implementing PRAISE for an introductory programming course are discussed. Each of these aspects is compared to previous evaluations where PRAISE was used in other disciplines.

This paper begins with a look at available peer-review systems. The PRAISE system is then described. In section 3 the method for evaluating the use of PRAISE is given. Results of this evaluation are shown in section 4. Other evaluations of peer review in computing science are shown in section 5 and related to the findings of this study. Finally, conclusions and recommendations are given in section 6.

2 Peer-review Systems

A number of peer-review systems are available. In this study we are most interested in systems that facilitate peer-to-peer evaluations of submitted documents, specifically programming assignments. The following sub-sections briefly introduce existing systems and compare them with PRAISE, the system used in this study.

2.1 Existing Peer Assessment Systems

A number of systems share commonalities with PRAISE (Chapman, 2006; Davies & Berrow, 1998; Hamer, Kell, & Spence, 2007; Kurhila, Miettinen, Nokelainen, Floreen, & Tirri, 2003). Examples include CPR, Aropä, and the Moodle Workshop Module.

The Calibrated Peer Review (CRP) system (Chapman, 2006) facilitates submission and review of essays. CPR requires students to undergo training to *calibrate* the peer reviews they later produce. Peer reviews created under CPR are subjective; by comparison PRAISE facilitates submission and review of documents in any format. PRAISE uses objective criteria and instructor moderation to ensure validity of marks without training students.

Aropä is a web-based peer assessment support tool that has been used in a range of academic disciplines (Hamer, Kell, &

Spence, 2007). Aropä allows students to upload documents of any format. Reviews are allocated manually or automatically by an instructor, following which students return to the system and are asked to give quantitative and qualitative feedback on a peer's submission. Quantitative feedback is governed by a flexible marking rubric. Reviews themselves can be subject to 'review' by instructors. Students are awarded marks based on an average of all peer reviews, with weightings given to reviews by instructors. PRAISE uses a similar system for submission and review but combines these two activities into a single step to minimise the number of visits required by students and eliminate complications of multiple deadlines. PRAISE aims at consensus from reviewers on objective binary criteria in order to determine marks. Where consensus is not reached, an instructor moderates the student's submission, overruling previous reviews.

The Moodle Workshop module allows students to submit any electronic document. Reviews can be allocated to students on an automatic basis. Reviews are based on a flexible marking rubric. Comments made by instructors can be saved and shared. The Moodle workshop module has multiple deadlines and does not allow for student flagging or moderation tracking (see section 2.2.2). Unfortunately this Moodle module has not been well maintained and is in a state of disuse within the Moodle community.

An automated peer-review add-on for the Coursemarker Programming Environment was described by Lewis and Davies (2004). Peer review can be combined with automatic assessment on a series of assignments. Peers select appropriate comments from a list; each comment carries a positive or negative mark which is awarded to the submitting student.

2.2 Description of the Peer-review System Used – PRAISE

PRAISE stands for Peer Review Assignments Increase Student Experience. Since its inception in 2004, PRAISE has been used in a computing concepts course offered to around 1000 non-computing students per year, a Masters level technology management course with approximately 140 enrolments annually, an introductory accounting course with 230 students, and a professional nursing course with 250 students. A modified version of PRAISE called SQLify is being developed for database courses with an emphasis on SQL query writing (Dekeyser, de Raadt, & Lee, 2007). PRAISE was first used with an introductory programming course in the second half of 2006.

PRAISE delivers rapid feedback to students from multiple sources. Details of the PRAISE system have been described previously with evaluations (de Raadt, Loch, & Addie, 2006; de Raadt, Toleman, & Watson, 2005, 2006). A brief description of PRAISE is given here to provide a context for the findings of this study.

Student	Instructor
	1. Create assignment instructions 2. Set review criteria
Beginning of Semester	
a. Create document b. Submit document c. Conduct reviews x 2 d. Receive peer feedback	3. Monitor Student Activity
Assignment Deadline	
e. Receive instructor feedback* f. Receive mark	4. Moderate reviews 5. Release Marks

Figure 1. Student interaction with PRAISE

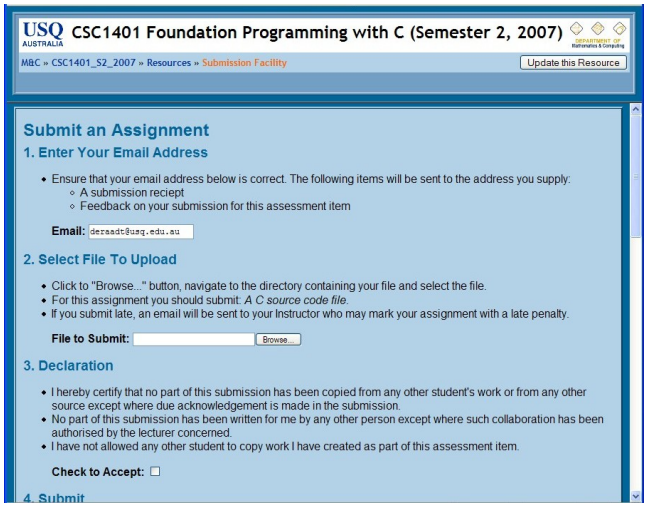


Figure 2a. Submission interface

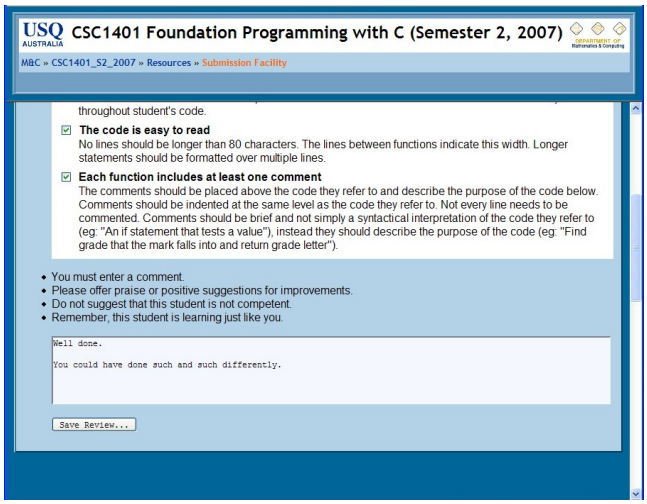


Figure 2b. Students check criteria and enter a comment

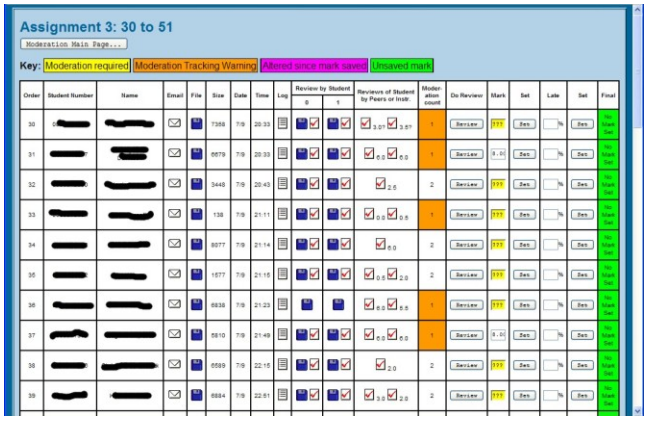


Figure 2c. Instructor's view of submissions

2.2.1 Process followed by a Student

For each assignment a student follows a process as described in Figure 1. Students read the assignment instructions and prepare their submissions as they would under a traditional assessment process. They may refer to the marking criteria

which are available prior to submission. When students have completed their document they submit it to the system (see Figure 2a). For the introductory programming course that is the focus of this study, source code files are submitted. The system verifies that the submitted file meets instructor-specified conditions such as type, size and content, and a receipt is emailed to the student.

Initially there is a pooling of submissions, but when this reaches a specified size (around 4-5) the system will begin to allocate reviews to students immediately as they submit their assignment. Students are then directed to complete reviews. The first students to submit must wait until the system notifies them by email to begin reviewing. A single submit-review step allows students to give reflective feedback immediately after submission and reduces the delay from submission to feedback receipt.

Students review the submissions of two peers and are rewarded with marks for undertaking reviews. For each review, students must download and open their peer's submitted document. Students complete a review by checking each criterion against their peer's submission (Figure 2b). Each criterion has a checkbox which is ticked if the peer has fulfilled the criterion. Criteria are phrased in a clear, objective fashion, so that students can review accurately even if they have failed to correctly achieve the criteria themselves. Criteria focus on completion of tasks rather than asking for a judgment of quality; this reduces ambiguity and increases consistency among reviewers. Students must give a comment; they are asked to give praise or positive suggestions for improvement. Students repeat this for each of the two reviews they conduct.

When students have completed reviews they wait to receive feedback from peers. An email is sent to students when their work has been reviewed by a peer. In their own time the students can view feedback on the system. Reviews are shown to students in the same web form used when they conduct reviews, but with controls disabled. Students do not know the value of each criterion and they will not see their overall assignment mark until it has been released by an instructor.

2.2.2 Process followed by an Instructor

Instructors follow a process for each assignment as shown on the right side of Figure 1. Before the semester begins the instructor must create the assignment. A key goal when using PRAISE is clear, objective criteria, focusing on completion of tasks set in the assignment instructions. Criteria should encourage consistency between reviewers. The criteria are stored in the system. Once this is done, students can begin the course and start completing assignments. Students can submit assignments at any time after the start of the course. Up to the assignment deadline the instructor will monitor the submission process but is not required to intervene.

Moderation of student reviews is achieved using the interface shown in Figure 2c. This interface shows a list of submissions for a particular assignment, each row relating to one student's submission. Instructors have access to each student's number, name, email address, submitted file, submission date, time and file size, a log of the submission details, the reviews conducted by the submitting student and peer reviews of the student's submission. Relationships between reviewer and reviewee are highlighted when the mouse pointer is moved over a review icon. The system attempts to consolidate reviews of the student's submission. If the submission has been reviewed twice and reviewers agree according to the criteria, the system will suggest a mark based on the value of each criterion. If reviews do not agree, the system will highlight the submission for instructor moderation. Past use of the system (de Raadt, Toleman, & Watson, 2005) indicates that the instructor will conduct moderations on roughly 50% of submissions depending on the complexity of the criteria; this means the instructor will accept a mark suggested by the PRAISE system, based on peer reviews, for 50% of submissions. This can allow time that would normally be spent marking to be used for other teaching activities. The instructor uses the same form that students use when conducting reviews. Students are notified by email when an instructor moderates their submission and the moderation appears with other reviews on the Marks and Reviews page.

When all submissions requiring moderation have been attended to and all conflicts are resolved, the instructor releases marks for all submissions of the assignment. Students are sent an email and can check their marks on the system.

2.2.3 Features

PRAISE boasts a number of features not available in other peer-review systems.

- **Single submit-review step**
  PRAISE can arrange new peer-review relationships for each assignment without instructor involvement. This is a big time-saver for instructors. This also benefits students. Only a single deadline is needed for both submission and review. Students are not required to return to the site for the sole purpose of completing reviews. Most students can immediately undertake reflection and evaluation on activities they have just completed. Waiting time to receipt of feedback is reduced. By allowing reviewing immediately after submission, students can work ahead in the course. In previous use of PRAISE some students have finished all the assignments of a course in the first few weeks. As students review previously submitted documents it is also easy to accommodate students submitting after the deadline. PRAISE applies late penalties automatically but late students can still complete reviews.
- **Practice submission**
  PRAISE allows only a single submission for each assignment. This can create anxiety in students unsure about using the system. To counter this, PRAISE can be set up with a 'practice' assignment allowing students to experience submission and review (with instructor-created documents to review).
- **Flagging**
  Even though instructors moderate assignments, some students are uncomfortable when peer reviews are used as a basis for creating marks. PRAISE allows students to flag peer reviews they believe are inaccurate. When a peer review is flagged an instructor must perform moderation on that student's submission.
- **Tracking moderations**
  An instructor can choose to award marks based on peer reviews when there is no conflict. Under this scheme, a top student who produces good work will consistently receive good peer reviews and may never receive a

moderation review from an instructor. If this is the case the student may feel they are not receiving the level of attention they deserve from the instructor. PRAISE counts instructor moderations for each student through the course. Targets can be set; for instance, "after assignment four all students have been moderated at least twice by an instructor." If the moderation count is below target, the instructor will conduct a moderation review, even if both peer reviews are consistent.

3 Evaluating Peer Review in an Introductory Programming Course

The following questions were used to guide the evaluation of peer review and of the PRAISE system in the context of an introductory programming course. An introductory programming instructor may ask these questions when considering adoption of peer review in their course.

RQ1. Can peer review be applied to assignments in an introductory programming course and what are the logistical differences when compared with a traditional submission model?

RQ2. Do novice programmers find PRAISE easy to use?

RQ3. Do novice programmers appreciate the learning benefits of undertaking peer review?

RQ4. Do novice programmers value reviews of their work by peers?

RQ5. Is there significant marking relief when using peer review compared to marking paper-based programming assignments?

Answers to these questions are considered in section 6.1 of the Conclusions.

3.1 Methodology

The use of PRAISE in an introductory programming course was evaluated in two ways. A survey, designed to elicit student attitudes towards the system, was conducted at two points during the course. Also, statistics on the use of the system by students were gathered from data stored in the system. This evaluation took place during the second semester (in the second half of the year) in 2007.

3.2 Setting – The Course[1]

The focus of this study is the use of peer review in an introductory programming course at the University of Southern Queensland. The course uses the language C in a procedural paradigm with a focus on syntax and sub-algorithmic problem-solving strategies.

Students are enrolled in the course in either *on-campus* mode or *external* mode. These two modes are distinguished by attendance, with on-campus students attending lectures, tutorials and practical classes. External students may be studying anywhere in the world. Based on first assignment submissions, 28% of students are enrolled on-campus and 72% externally.

There are six assignments in the course, each requiring the novice programmer to generate a source code file containing a problem solution. Each assignment contributes 8 marks to the final assessment; the remaining 52 marks are allocated to the end-of-course examination. Within each assignment 6 marks are allocated to the quality of the student's submission as judged through peer reviews and instructor moderation. A further 2 marks are awarded for completing two reviews (one mark per review). To evaluate code submitted by peers, students are asked to compile, run and test the solutions while checking the review criteria. This form of testing, as part of reviews, has not formerly been used with PRAISE.

Assignment deadlines are regular, roughly two weeks apart. Assignment deadlines occur at midnight on the due date. After this, late penalties are applied to encourage students to stay on track. Smaller, regular assignments are used to encourage continuous involvement in the course. Students must complete one assignment before they can move onto the next. Regular assignments allow easy identification of students falling behind, who might require intervention.

Support mechanisms provided to students include online forums, email, phone and personal contact with instructors. Students are encouraged to make use of the support mechanisms in that order unless personal matters arise. The forums are monitored on a regular basis.

Information was provided to students explaining why peer review is used in the course. Students were able to read a justification for using the system, view a short video of how to use PRAISE and try out the system through a practice assignment.

3.3 Survey

Two surveys were conducted during the semester. Students were able to complete the first survey after finishing the first assignment and the second after the sixth (and final) assignment. The surveys were conducted online using a web form. The questions used in the survey were drawn primarily from previous evaluations of the system (de Raadt, Toleman, & Watson, 2005, 2006). Questions used the statements in Table 1. In the second instance of the survey, one question was dropped (question 2) and several were added. Questions used with each survey are marked with a tick in the survey column of Table 1.

Questions asked in the first survey at the beginning of the semester (after the first assignment) were phrased in the present tense. Questions in the second survey asked at the end of the course reflected back on the use of the system using the past tense. For instance, "there is support" was later phrased "there was support" and "seems easy to follow" was later "seemed easy to follow". The subject of each question was the same in both surveys.

The questions focused on the course, the assignments used in the course, and, of primary concern in this study, students' attitudes towards peer reviewing. Students were asked for their agreement with the statements in Table 1 and responses were captured using a five-point Likert scale with possible responses *Strongly Disagree*, *Disagree*, *Neutral*, *Agree* and *Strongly Agree*. Negatively phrased statements are marked with an asterisk (*).

[1] The term course is used to refer to a single semester-long period of study. This may be equivalent to a subject, unit or paper in other institutions.

Table 1. Survey questions for both surveys

	Question	Survey 1	Survey 2
1	I feel confident that I will pass this course.	✓	✓
2	This course is important to my degree program.	✓	
3	I enjoyed the challenge of completing programming activities in this course.	✓	✓
4	The assignments were big and took a lot of time to complete.*	✓	✓
5	There was support if I got stuck when completing assignments or reviews.	✓	✓
6	The process of submitting assignments was easy to follow.	✓	✓
7	The process of completing reviews was easy to follow.	✓	✓
8	I felt limited by only being able to submit each assignment once.*		✓
9	Submitting assignments electronically requires less effort than submitting an assignment on paper.		✓
10	Completing regular assignments forced me into a regular pattern of study.		✓
11	Reviewing other's work helped me understand the concepts covered in each assignment.	✓	✓
12	Seeing the work of others showed me different ways to complete tasks.	✓	✓
13	I would rather receive marks from instructors only.*	✓	✓
14	Interacting with peers through reviewing motivated me to produce better assignments.	✓	✓
15	Communicating with peers through reviewing gave me the sense I was not alone in my studies.	✓	✓
16	I was uncomfortable that others saw my work.*	✓	✓
17	When I saw other students' submissions I compared them to my own work.		✓
18	The feedback I received from my peers through reviews was useful to me.		✓
19	Feedback on my submissions came rapidly from peers and instructors.		✓
20	The quality of feedback from peers and instructors was as good or better than what I would expect on paper based assignments marked by hand.		✓
21	I would be happy to use the same submission and review facilities in other courses.		✓

3.4 Usage Statistics

A number of statistical measurements of the system were achieved by analysing data available in the system. The aspects measured were as follows.

- Time from submission to deadline
- Time from submission to receipt of first review
- Proportion of moderations required due to conflicts
- Use of flagging by students

4 Results and Discussion

This section discusses the results of the evaluation. First the survey participants' responses are described. Following this, usage statistics are shown. Finally, the results are compared with previous evaluations of the use of PRAISE.

4.1 Survey Responses

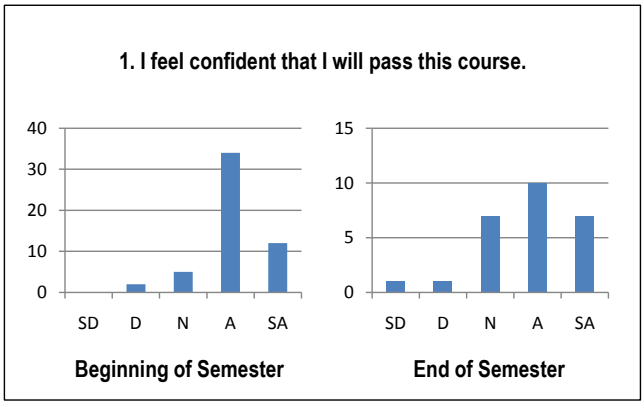
Table 2 shows response rates for the two surveys. Participants include males and females, school leavers and mature-age students and part-time and full-time students. The proportions for these aspects were not captured as part of the survey.

Table 2. Survey response rates

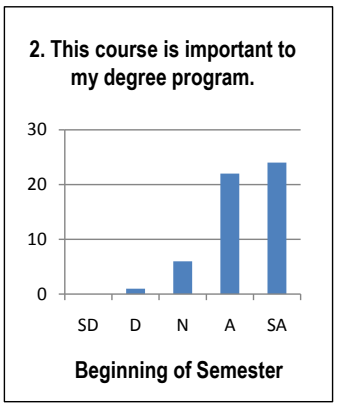
	Submissions	Surveys	Response Rate
Asst. 1/Survey 1	79	53	67%
Asst. 6/Survey 2	38	26	68%

There were 14 participants who responded to both the first and second surveys. The responses to each survey are considered independently rather than as a continuous change of attitude. Responses are grouped by the focus areas: course, assignments and reviewing.

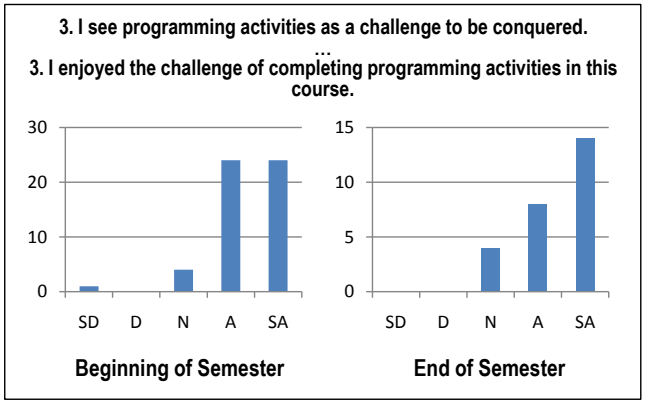
4.1.1 Questions about the Course



When asked if they were confident about passing the course almost all students showed confidence at the beginning of the semester (question 1 beg: SD+D=4%,N=9%,A+SA=87%). Closer to the end of the semester, students were predominantly confident, but some gave a neutral response (question 1 end: SD+D=8%, N=27%, A+SA=65%).



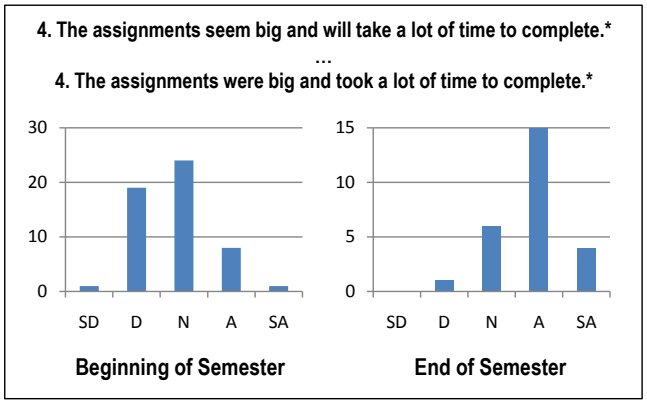
At the beginning of the semester, most students suggested the course was important to their studies (question 2 beg: SD+D=2%, N=11%, A+SA=87%).



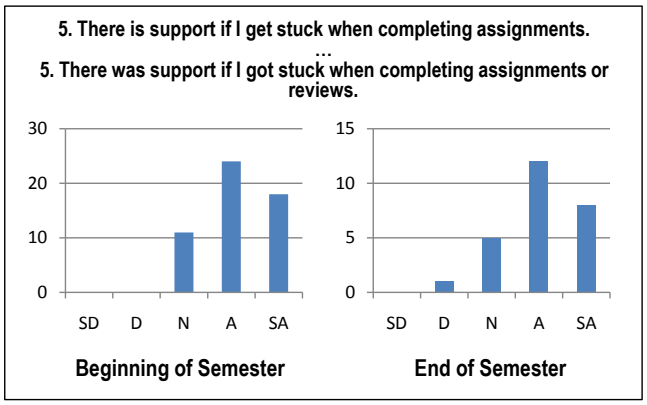
At the beginning of the semester, most students agreed that programming is challenging (question 3 beg: SD+D=2%, N=8%, A+SA=91%). Responses to these questions paint a positive picture for the course. Participating students seem to have good intentions and motivation.

At the end of the course students were asked if they enjoyed the challenge of programming and another strong response was recorded (question 3 end: SD+D=0%,N=15%,A+SA=85%).

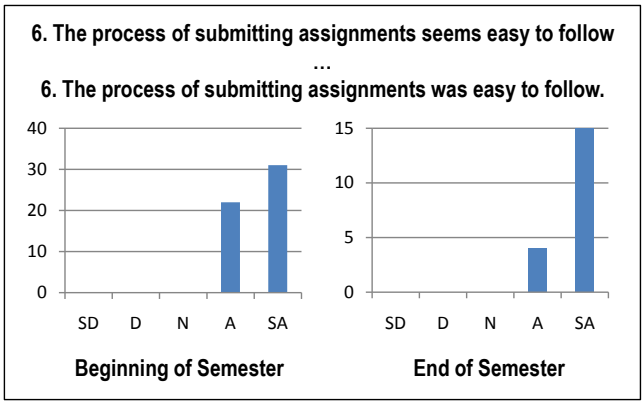
4.1.2 Questions about the Assignments

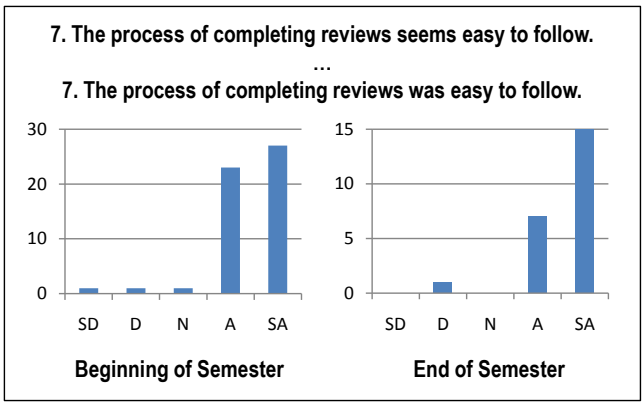


Students were divided over their perceptions of the scale of the assignments at the beginning of the course. Question 4 was phrased negatively and yielded responses (question 4 beg: SD+D=38%, N=45%, A+SA=17%) revealing many neutral participants. At the end of the semester, after doing the work, more students agreed that the assignments were big (question 4 end: SD+D=4%,N=23%, A+SA=73%).

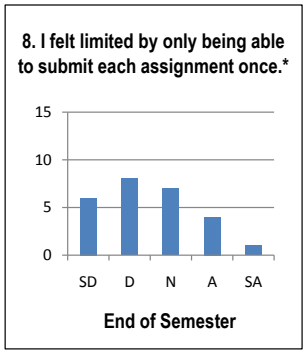


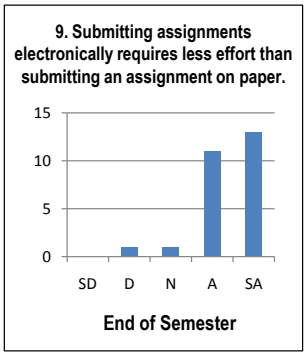
Students seem happy with the apparent level of support as shown from responses to question 5 (beg: SD+D=0%, N=21%, A+SA=79%; end: SD+D=4%, N=19%, A+SA=77%).

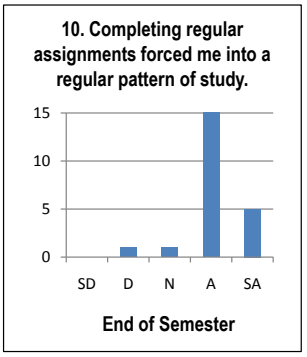




Questions 6 and 7 relate to the ease of submission (beg and end: SD+D=0%,N=0%,A+SA=100%) and review (beg: SD=4%, N=2%, A+SA=94%; end: SD=4%, N=0%, A+SA=96%). Both early in the course and at the end, students seem very at ease with both these processes.





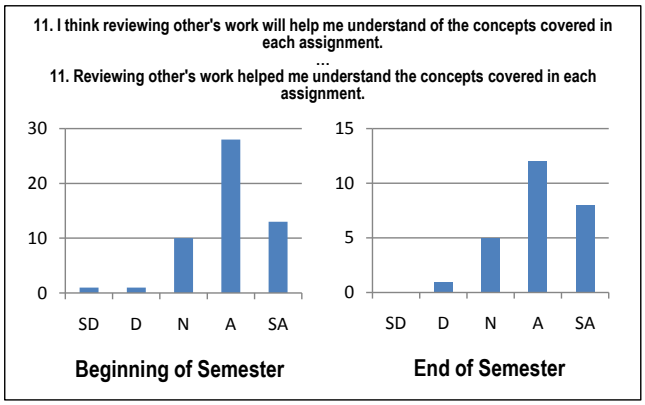


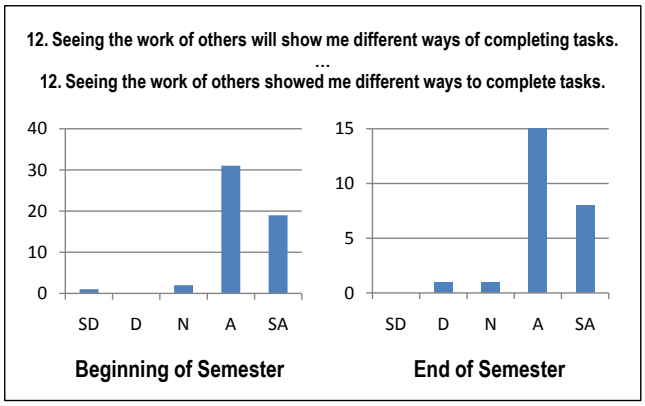
Questions 8, 9 and 10 were asked only at the end of the semester. Question 8 related to the limitation of only being able to submit once for each assignment. This was a negatively phrased question showing responses (question 8 end: SD=54%, N=27%, A+SA=19%). These responses indicate that a majority of students are comfortable with the single submission but there is a large number who are not. Question 9 asks about the ease of submitting an electronic document over a paper submission (question 9 end: SD=4%, N=4%, A+SA=92%). This finding is useful even for instructors using

electronic submission without peer review. One of the intentions for having six regular assignments was to maintain regular student involvement in the course. Students agreed that this had been achieved (question 10 end: SD=4%, N=4%, A+SA=92%).

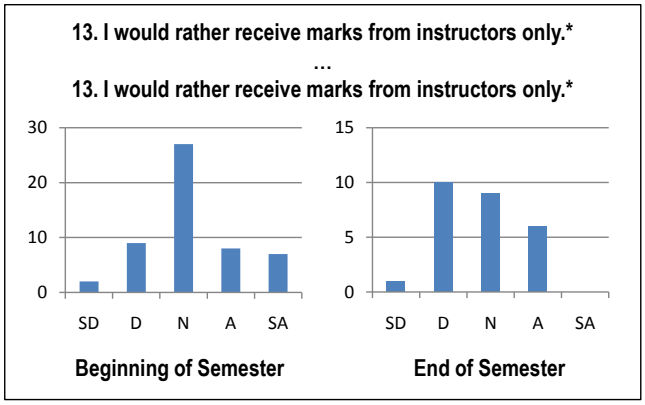
4.1.3 Questions about Reviewing

Questions 11 to 21 were designed to discovered how students value reviewing as part of their assessment.

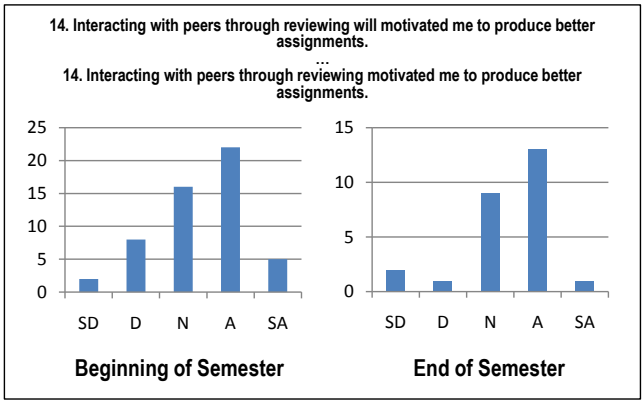




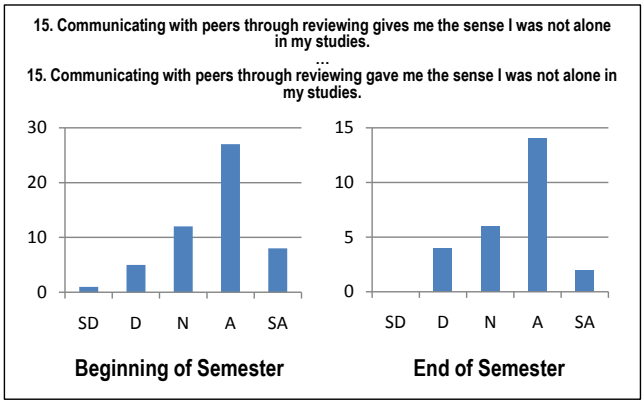
Results for question 11 (beg and end: SD+D=4%, N=19%, A+SA=77%) and question 12 (beg: SD+D=2%, N=4%, A+SA=94%; end: SD+D=4%, N=4%,A+SA=92%) describe the participating students' perceptions of the learning benefits inherent in undertaking peer review. It is clear that students saw these benefits early in the course and at the end. Programming students really appreciate seeing the solutions submitted by their peers.



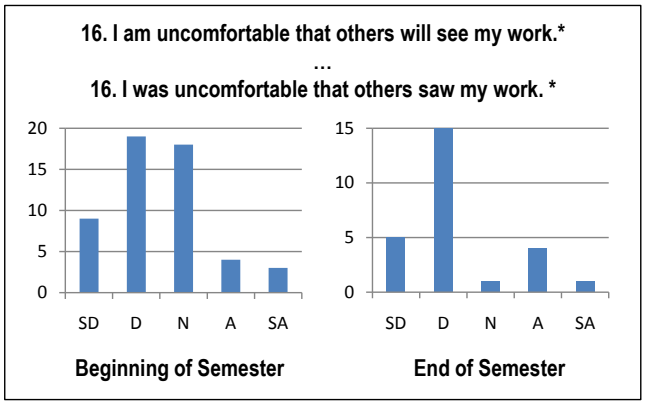
Question 13 puts a value on the use of peer reviews as a means of assessment (question 13 beg: SD+D=21%, N=51%, A+SA=28%; end: SD+D=42%, N=35%, A+SA=23%). This question is phrased negatively. At the beginning of the semester most students were neutral in their response, but it is clear that a good proportion of students want an authoritative instructor awarding marks. At the end of the semester students seemed to value peer-review slightly higher. This implicitly gives a value to the feedback students receive from their peers. It should not be assumed that feedback in peer reviews is valued as highly as instructor feedback.



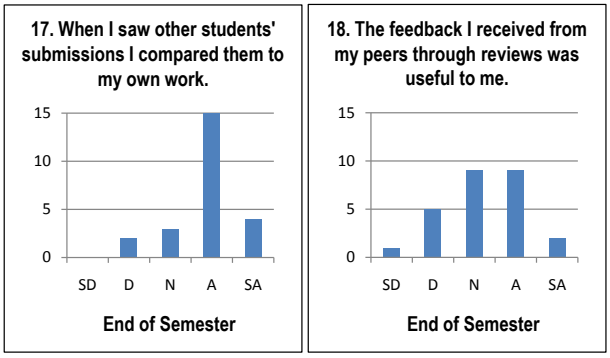
Question 14 measures the motivation of participating students gained by knowing that a peer will see their submission (question 14 beg: SD+D=19%, N=30%, A+SA=51%; end SD+D=12%, N=35%, A+SA=54%). Many students feel motivated by this (more than in any previous cohort). A few students do not, but this does not necessarily imply that peer reviewing is de-motivating; it may be that participating students who disagreed with this statement are motivated by forces other than their peers seeing their work.



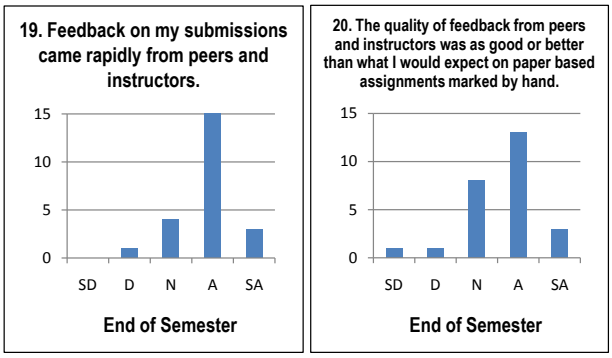
Question 15 measures the sense of community that arises out of peer review (question 15 beg: SD+D=11%, N=23%, A+SA=66%; end: SD+D=15%, N=23%, A+SA=62%). As mentioned earlier, many students in the course are externals who can feel isolated in their studies. It appears that, for most students, peer reviewing encourages a sense of community which, together with online communication, can positively affect learning outcomes.



Question 16 measures the level of comfort with peers viewing a student's submission. The system provides double-blind anonymity in reviews. This question was phrased negatively with early responses (question 16 beg: SD+D=53%, N=34%, A+SA=13%) suggesting students are mostly comfortable or neutral. At the end of the semester most students suggested they were comfortable (question 16 end: SD+D=77%, N=4%, A+SA=19%). Few students are uncomfortable, but this is still an area of concern.

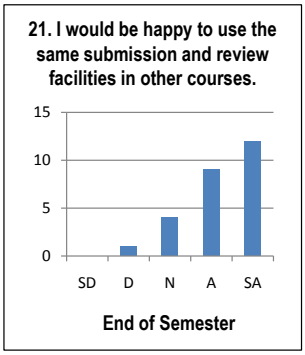


Questions 17 to 21 were asked in the end of semester survey only. Responses to question 17 (end: SD+D=8%, N=12%, A+SA=81%) indicate students had undertaken reflection, which is one of the desired pedagogical benefits of peer review. Responses to question 18 are mixed (question 18 end: SD+D=23%, N=35%, A+SA=42%) and show what has been found in previous surveys from other disciplines: that students do not necessarily value the feedback they receive from peers.



Question 19 describes the students' satisfaction with the speed at which they received feedback and this is quite positive (question 19 end: SD+D=4%, N=15%, A+SA=81%).

Question 20 asks if students are happy with the general quality of feedback they receive through the PRAISE system when compared to paper-based assignments. Students are generally happy with the quality of feedback they received (question 20 end: SD+D=8%, N=31%, A+SA=62%).



The final question, question 21, asks if students would like to use a system like PRAISE in other courses they are studying. Students were quite happy with the system and would like to see it used elsewhere (question 21 end: SD+D=4%, N=15%, A+SA=81%).

4.2 Comments

The free comments made by students were encouraging and predominantly positive. The following are positive comments from the initial survey.

- *I think reviewing other peoples work is great.*
- *…the support available is fantastic.*
- *i like the review system, it works well.*
- *I like the regular assigments and the review system.*
- *I was worried at first that other students would be able to view my work. However, since there are strict guidelines as to how to review someones work and that we are all encouraged to give positive feedback, I felt more comfortable.*

After the initial survey, one student raised a problem unique to using peer review in a programming context where students need to compile and test code. Students are encouraged to use an ANSI standard compiler. Examples are suggested to students and a free compiler is available for download. Students are asked to be aware that peers using other compilers may be reviewing their work. However, there is never complete compatibility between different compilers and how they behave.

- *…when I compiled my assignment through OSX terminal with g++, I got no warnings or errors, but people on windows compiling my source code did.*

Initially, one student stated their refusal to undertake peer reviews.

- *I have great reservations with "peer review", and for myslf do not partake due to the possible harm caused. After all what would a student know about the subject they are learning? … I would prefer information straight from the lecturer as I would trust the souce of the information…*

This student left their name with this comment. This was seen as an invitation for a response. The student was encouraged to undertake reviews as a learning activity for their own benefit and assured that the process of reviewing is was overseen by instructors. The student did go back and complete reviews. This comment exemplifies a nervousness about the use of peer review for assessment, which itself is quite novel. Students must be made aware of the justification and learning benefits of peer reviewing before they are involved.

In the second survey responses were still predominantly positive.

- *...set up brilliantly to study externally…*
- *Peer review is a new experiecne for me, an uncomfortable one at 1st, but it can be of benefit if the student puts in the work to start with*
- *I really liked the fact that there was so much flexability…*
- *I found the assignments to[o] big but highly beneficial. It allowed me to correct my own mistakes and write better code…*
- *I enjoyed this course and was challenged by it. Mostly I found the comments from peers to be supportive and helpful…*

After experiencing the system over the semester, a number of students raised their concerns about certain aspects of the system, many related to workload in the course.

- *…I often did not know what to say because if their code was not working I had no clue why.*
- *I strongly feel that this unit covers far too much material over a short period of time.*
- *...six assignments may have been a little over the top, however it did make me study more regularly.*
- *The regimented structure of assignment submission dates for this course is hard for students studying and working full-time ... I found myself in the position of attempting assignments without having done the required course work*
- *...only negative i found was review comments weren't particularly useful.*

4.3 Usage Statistics

Statistics were gathered regarding timing, reviews and moderation.

4.3.1 Time from submission to deadline

PRAISE allows students to work ahead. Students are made aware of this fact at the start of semester. Some students take advantage of this, others do not, but neither is necessarily preferred. However, measuring how far ahead students are working is an indication of student motivation. Table 3 shows statistics about the time between submission and the deadline. Late submissions are excluded as these may have involved extensions or other complicating factors. The median is the best guide and shows a reduction, with half of the student cohort submitting assignment 1 one day and nine hours before the deadline but only three to six hours before the deadline in the last three assignments.

Table 3. Time between submission and deadline

Asst.	Longest (Earliest)	Mean	Median	Shortest (Latest)
1	23days	3days 3hr	1day 9hr	60min
2	13days	1day 17hr	12hr 20min	16min
3	5days	1day 1hr	11hr 9min	2min
4	9days	1day 12hr	6hr 26min	5min
5	2days	8hr	2hr 50min	0min
6	9days	19hr	5hr 42min	3min

4.3.2 Time from submission to receipt of first review

One of the benefits of a single submit-review step is that students can receive reviews from peers shortly after they submit. To measure the effectiveness of this feature the delay between a submission and the first review is captured. These figures also give an indication of the time taken by students to complete reviews. The longest and shortest delays, and the mean and median delays are shown in Table 4.

Table 4. Time from submission to first review

Asst.	Longest	Mean	Median	Shortest
1	13days	15hr	4hr 8min	11min
2	10days	1day 4hr	2hr 45min	24min
3	3days	8hr	3hr 18min	12min
4	7days	23hr	2hr 9min	10min
5	3days	5hr	1hr 6min	25min
6	9days	20hr	1hr 47min	31min

Again the best guide to measuring the delay for feedback is the median delay. For the first assignment, half of the student cohort received feedback within 4 hours or less. For later assignments this reduced to a little over an hour. These delays are affected by the time between submission and the deadline. When students submit closer to the deadline there is a greater concentration of submissions, so feedback is returned sooner. Students submitting earlier (further from the due date) generally have to wait longer for feedback to arrive. All of these figures, though, are commendable considering that the delay from submission to feedback receipt in a traditional paper-based assessment involving postage can be up to six weeks.

Also of interest in Table 4 is the minimum time between submission and review. Although students were generally receiving feedback faster over the semester, the minimum gap increased in the later assignments, showing that students were taking more time to complete reviews, perhaps due to the greater complexity and number of review criteria.

4.3.3 Proportion of moderations required

The proportion of submissions which require moderation is a measure of the consistency achieved between peer reviewers. This in turn is an indication of the ease with which students were able to apply the review criteria. The rates where moderation was required are shown in Table 5.

Table 5. Proportion of moderation required

Assignment 1	61% (100% conducted)
Assignment 2	72%
Assignment 3	80%
Assignment 4	76%
Assignment 5	67%
Assignment 6	73%

For the first assignment all submissions were moderated, even though only 61% of reviews were conflicting. This was done to encourage students early and provide a good example of the reviewing standard expected. It also provided a chance to detect lazy reviewers – students who simply check all criteria without referring to, or testing, the submitted source code. For

later assignments, the number of conflicts, and therefore moderations, increased. It should be noted that there were more criteria used with reviews in these later assignments, which may have increased the likelihood of conflicts.

4.3.4 Proportion of Reviews Flagged

The last measure gathered from use of the system was the proportion of peer reviews flagged by students. If students were unhappy about a review they had the option of flagging it. A flagged review forces an instructor to moderate the submission. The level of flagging for the assignments is shown in Table 6.

Table 6. Proportion of all peer reviews flagged

Assignment 1	3%
Assignment 2	4%
Assignment 3	2%
Assignment 4	2%
Assignment 5	1%
Assignment 6	3%

The level of flagging is an indication of the confidence students place in the reviews they receive from their peers. From the survey questions described earlier it is clear that, while students value the experience of undertaking reviews, they do not always have confidence in the feedback they receive from peers. Despite this, the use of flagging was quite low, indicating that students either believe the reviews are accurate or are confident an instructor will correct inaccurate reviews.

4.4 Comparison to Previous Evaluations in non-Programming Courses

Novice programmers find submitting assignments and conducting reviews easy. Their confidence is superior to students from other disciplines in previous evaluations.

Previous evaluations have shown that students do not value reviews from peers as highly as instructor feedback. This attitude is also evident in the current evaluation, with novice programmers valuing peer reviews slightly less than in previous evaluations (see question 13).

Students participating in the current evaluation are more motivated than students in previous evaluations by knowing peers would view their work. Survey participants in previous evaluations were relatively neutral about viewing and evaluating the work of their peers. A clear distinction to previous evaluations is the high value students place on being able to view, test and evaluate the work of peer novices. In introductory programming this appears to be a major attraction.

Students gave enthusiastic comments about seeing others' work and showing off their own work. Some negative attitudes were given in comments. Most negative comments were based on the workload of the course rather than the use of peer review. In previous evaluations it was concluded that many negative attitudes arose from students being ill-informed about the motivation for using peer review and unaware of the benefits to learning outcomes. The response has been to promote peer review and its benefits prior to use. This was done in the current course but perhaps this dissemination could be improved. Several students felt the assignments were too big.

Rates of moderation were higher than experienced in other disciplines through previous evaluations. This indicates that students are producing less consistent reviews, which is a sign of the quality and complexity of the assignment instructions and criteria. Clear criteria need to be created and refined, which may require several iterations of each assignment.

Previous evaluations found that most students submit on the due date, but in each course where evaluation was undertaken several students would work ahead, some completing all assessments in the first few weeks. This does not seem to be the case in the introductory programming context. Novice programmers submitted closer to the due date and no student worked to submit assignments ahead of schedule, even after they were encouraged to do so.

Novice programmers took more time to produce reviews than has been experienced in other disciplines. In a computing concepts course for non-computing students, the median time from submission to first feedback was 1hr 21min where in the current course the overall median was 2hr 33min. Novice programmers took longer to evaluate the work of their peers. Survey participants indicated that they enjoyed seeing the work of their peers and comparing it to their own.

5 Relation to Previous Evaluations of Peer Assessment in Programming

Sitthiworachart and Joy (2004) describe the use of peer review together with automatic marking for a single assignment in an undergraduate programming course. The workings of the system used are not described in detail, however some information is given. For the peer-review component, students are asked to subjectively rate three peers' submissions using set criteria, each associated with a scale of marks. Marks awarded to students are an average of three reviews of their submitted work. In evaluating their system Sitthiworachart and Joy found 65% of students were satisfied with their marks and 51% regarded feedback from peers as useful. Through a combination of attitudinal measures captured in this study it could be argued that student satisfaction with PRAISE is higher, but it is interesting to note that peer feedback was not valued highly in either evaluation. Students expressed a lack of confidence in their marks in comments under the system used by Sitthiworachart and Joy, which caused them to suggest moderation as a means of providing fairer reviews. PRAISE uses instructor moderation, which may be why students showed higher confidence in that system.

A study by Chinn (2005) measured the validity of peer-assessment in an algorithms course. The study found a correlation between marks from peer-reviewed and other activities, suggesting that peer-awarded marks are consistent. Chinn noted that students tend to focus on high-level errors, identifying these more often than low-level errors. Student attitudes towards the validity of peer assessment discovered by this study indicate that novice programmers accept the marks they receive, with relatively low levels of flagging.

6 Conclusions

In this section the questions raised in section 3 will be addressed first. This is followed by discussion of differences encountered between use of PRAISE in an introductory

programming course and in other courses. Finally, future work is suggested.

6.1 Research Questions

RQ1. Can peer review be applied to assignments in an introductory programming course and what are the logistical differences when compared with a traditional submission model?

Peer review fitted the assignments in the introductory programming context nicely. Using simple, fixed criteria it was possible to focus student attention on important syntactical and problem-solving aspects of assignments. Peer review has allowed for smaller, more frequent assignments focused on recent topics.

One difference comes in asking students to undertake testing of their peers' solutions for reviews. Previous use of PRAISE has asked students to undertake relatively passive observations when evaluating the work of their peers.

Anonymity becomes a difficult balancing act when using peer review. In examples shown to novices, comments are written at the start of source code files to identify the author and other relevant details. Such comments are encouraged in the course, but students have to be asked to remove these comments before submitting and many fail to do so. Some aspects of the assignments are designed to allow students to personalise their work, hopefully making the tasks more relevant to them. An example of this occurs in most assignments. One example in the first assignment involves students outputting their name in asterisks. While these aspects of personalisation are pedagogically desirable, they reduce the level of anonymity and potentially the accuracy of reviews if peers are familiar with each other.

Some compatibility issues arose during reviews of assignments. Students are working on different platforms and development environments so while one compiler might not warn a novice to add a blank line at the end of their source code, another compiler will. Asking students to consider the environment where their code will be tested is not bad as it encourages them to write more compatible code and avoid compiler specific tricks.

RQ2. Do novice programmers find PRAISE easy to use?

Absolutely, and with more confidence than any previously surveyed cohort of PRAISE users from other disciplines.

RQ3. Do novice programmers appreciate the learning benefits of undertaking peer review?

Previous studies have shown that peer review encourages students to become more involved in the course, to feel less isolated, and to move towards higher-order thinking

It appears that novice programmers recognise the benefits of conducting peer reviews. They relish the chance to view, test and evaluate the code of others. Many are motivated to produce better work because of peer review.

RQ4. Do novice programmers value reviews of their work from peers?

Novice programmers are quite neutral about whether they would prefer feedback from peers and instructors through reviews or from instructors alone. Some students feel this is beneficial and some feel it could be detrimental. Additional feedback to students should positively improve their learning outcomes, but even without this extra feedback, the remaining benefits inherent in peer review still make its use worthwhile.

RQ5. Is there significant marking relief when using peer review compared to marking paper-based programming assignments?

There is some reduction in the marking of individual assignments. The process of moderation is somewhat quicker than marking code on paper or by other means. The number of submissions marked by an instructor can be reduced by 20-40% in an introductory programming course using peer review as a basis for assessment. This is not as significant as in other disciplines. Perhaps this rate of moderation can be improved by refining review criteria.

There are costs associated with establishing good criteria and managing the system, but then there may be equivalent costs in any submission and marking system.

Based on the answers to these research questions the authors recommend that introductory programming instructors considering adopting peer review to improve learning outcomes for their students.

6.2 Comparison with Non-programming Courses

A number of differences were found between the attitudes and practices of novice programmers undertaking peer review and those of students from other disciplines. The following is speculation on why these differences are occurring.

Why are novice programmers more motivated by peer review?

Assignments in introductory programming courses are arguably more challenging than in those in other disciplines. Assignments require students to undertake problem solving, and solutions are students' expressions of their development in programming expertise. Peer review gives novice programmers an opportunity to showcase their achievements.

Why don't novice programmers work ahead?

It seems likely that novice programmers would work ahead if they could. It may be they are prevented from doing so by the cumulative nature of materials which build up over the course of study. It may be that programming concepts require longer to absorb. It may be that assignments are more challenging than assessments in other disciplines, taking longer to produce submissions. Then again it may be that novice programmers, or perhaps just this cohort, are less motivated to work ahead.

Why do students take longer to conduct reviews?

One reason novices take longer in reviewing may be that they are asked to compile, run and tests their peers' solutions, taking more time than would be needed to simply read and evaluate a submission. Another reason may arise from students finding the work of their peers more valuable in novice programming than in other disciplines, and therefore spending more time observing the techniques and methods applied by their peers.

6.3 Future Work

In future semesters the findings of this evaluation will be used to improve the assignments and criteria used for peer review. Re-evaluation will be undertaken to measure any improvement.

The creators of PRAISE want to share the PRAISE system more widely with instructors. One possible avenue being pursued is to assist in improving the Moodle Workshop module which is languishing. Reinforcing the value of reviews by assessing the quality of reviews provided by students is another aspect of future development and investigation.

7 References

Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., et al. (Eds.). (2001): *A Taxonomy for Learning, Teaching and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives*. New York, USA: Addison Wesley Longman, Inc.

Brook, C., & Oliver, R. (2003): Online learning communities: Investigating a design framework. *Australian Journal of Educational Technology,* **19**(2):139 - 160.

The White Paper: A Description of CPR, Chapman, O. L. http://cpr.molsci.ucla.edu/cpr/resources/documents/misc/CPR_White_Paper.pdf Accessed February 23 2006.

Chinn, D. (Year): Peer assessment in the algorithms course. *Proc. Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiSCE2005)* 69 - 73.

Davies, R., & Berrow, T. (1998): An evaluation of the use of computer supported peer review for developing higher level skills. *Computers Educ,* **30**(1/2):111 - 115.

Nomination Statement - Awards for Programs that Enhance Learning, de Raadt, M., Loch, B., & Addie, R. http://www.sci.usq.edu.au/staff/deraadt/award/. Accessed 13th September 2007.

de Raadt, M., Toleman, M., & Watson, R. (Year): Electronic peer review: A large cohort teaching themselves? *Proc. Proceedings of the 22nd Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE'05)*, Brisbane 159 - 168, QUT, Brisbane.

de Raadt, M., Toleman, M., & Watson, R. (2006): An Effective System for Electronic Peer Review. *International Journal of Business and Management Education,* **13**(9):48 - 62.

Dekeyser, S., de Raadt, M., & Lee, T. Y. (Year): Computer Assisted Assessment of SQL Query Skills. *Proc. Eighteenth Australasian Database Conference (ADC 2007)*, Ballarat, Australia 63:53 - 62, ACS.

Hamer, J., Kell, C., & Spence, F. (Year): Peer assessment using aropä. *Proc. Proceedings of the ninth Australasian conference on Computing education (ACE2007)*, Ballarat, Victoria, Australia 43 - 54, Australian Computer Society, Inc.

Kurhila, J., Miettinen, M., Nokelainen, P., Floreen, P., & Tirri, H. (Year): Peer-to-Peer Learning with Open-Ended Writable Web. *Proc. Proceedings of the 8th annual conference on Innovation and technology in computer science education (ITiCSE '03)*, Thessaloniki, Greece 173 - 178, ACM Press.

Lewis, S., & Davies, P. (Year): Automated peer-assisted assessment of programming skills. *Proc. Second International Conference on Information Technology: Research and Education (ITRE 2004)* 84 - 86.

Prins, F. J., Sluijsmans, D. M. A., Kirschner, P. A., & Strijbos, J.-W. (2005): Formative peer assessment in a CSCL environment: a case study. *Assessment & Evaluation in Higher Education,* **30**(4):417 - 444.

Saunders, D. (1992): Peer tutoring in higher education. *Studies in Higher Education,* **17**(2):211 - 218.

Sitthiworachart, J., & Joy, M. (Year): Effective peer assessment for learning computer programming. *Proc. Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education (ITiSCE2004)*, Leeds, United Kingdom 122 - 126.