

:: Refereed Article A6:**Towards a framework for a code review process**

Xiaosong Li
Unitec, New Zealand
xli@unitec.ac.nz

Donald Joyce
Unitec, New Zealand
djoyce@unitec.ac.nz

Li, X. (2008). Towards a framework for a code review process. *Journal of Applied Computing and Information Technology*, 12(1). Retrieved June 2, 2015 from http://www.citrenz.ac.nz/jacit/JACIT1201/2008Li_CodeReview.html

Abstract

This paper describes a longitudinal study of a Code Review Process (CRP) which used an action research method. The CRP was used as one of the assessment methods in a third year (Level 7) undergraduate Web Application Development (WAD) course. This paper reviews the past three cycles of the study in order to get a deeper understanding of the issues. To address the issues more effectively and better meet the students' needs, the existing CRP is refined to develop a code review framework. The paper discusses different options and proposes a framework consisting of five components. To implement the framework, the course assessment scheme needs to be redesigned. Initially, the framework should be implemented in the same course. If successful, there is a potential to revise the framework and introduce it into similar courses at Levels 5, 6 and 8.

h3

Introduction

This paper describes a CRP employed by the author as one of the assessment methods in a third year (Level 7) undergraduate WAD course. The purpose of the CRP was to motivate students to apply coding standards and to facilitate the communication amongst the students. It was proposed by Li and Prasad (2005) as a strategy for teaching coding standards. It is common practice in industry to implement coding standards by using a CRP in which a team of developers check and make suggestions on the code produced by one or more members in the team according to formal coding standards (Dimon, 2006; Parasoft, 2005, 2007; Pfeiffer, 2005). Code reviews have many purposes or benefits (Bogue, 2006), these can be summarised into three categories:

- to help developers identify problems and envision new solutions
- to help developers learn when and how to apply techniques to improve code quality, consistency, and maintainability.
- to help developers get deeper understanding about the code under development.

Using the CRP in the classroom should equip the students with the code review skills needed in their future workplace. The CRP is a type of peer review (Carlson, Berry, & Voltmer, 2005; McGourty, Dominick, & Reilly, 1998; Trahasch, 2004) and also provides an opportunity for the students to learn collaborative learning skills.

The CRP has been used in the WAD course as one of the routine assessment instruments since Semester 2, 2005. Li (2006) analysed the results of the initial implementation. In Semester 2, 2006, three improvements were introduced. Li (2007) compared the results from 2005 and 2006 and proposed a set of strategies for future improvements. This paper analyses the additional observations in Semesters 1 and 2 of 2007, extends the results of Li (2007), and proposes a code review framework for future and wider usage.

Methodology

This study of the CRP uses an action research process. In action research, initially, a problem is identified and data are collected for more detailed analysis. This is followed by postulation of several possible solutions, from which one is chosen for implementing. More data are collected and analysed, and the findings are interpreted to determine how successful the implementation has been. At this point, the problem is re-assessed and the process begins another cycle. This process continues until the problem is resolved (O'Brien, 2001).

This study consists of three cycles. A trend study survey was used as one of the instruments to monitor the CRP after the first and second cycles of the study. A trend study samples different groups of people at different points in time from the same population. The trend studies are uniquely appropriate for assessing change over time (O'Connor, 2007; Web Centre for Social Research Method, 2007). In both surveys seven identical rating scale questions were used. However, these quantitative data were not obtained for the purpose of carrying out rigorous statistical analysis, but to provide sufficient qualitative data to investigate the most suitable strategies for future teaching practice (Weinreich, 1996). The data from the two surveys were interpreted using an "interpretive" approach in order to present a holistic view of data rather than a condensed view (Savenye & Robinson, 2003). The aim was to get some evidence as to whether the improvements introduced in the second cycle worked and what further improvements should be made.

In addition to the surveys, the lecturer's observations, an informal interview, the lecturer's informal conversations with the students, and the literature were also used to monitor and analyse the CRP process. The electronic survey was not continued in the third cycle; instead, the lecturer's observations were used as the main instrument to monitor the CRP. One of the reasons for this was that no significant change was introduced to the CRP in the third cycle. So the analysis of this cycle is more about finding solutions for the existing problems rather than checking if any change worked. Another reason was that the student numbers were getting lower, so the sample sizes were getting smaller and smaller.

Information About Course and Students

The purpose of the WAD course is to enable students to gain knowledge and develop skills for developing client-server and web-based applications in the Intranet/Internet environment. The CRP has been used for the first assignment of the WAD course for three years. The assignment required the students to design and implement a token web application by using ASP.NET/VB.NET. A large amount of VB.NET code was involved. The focus of this course is software development rather than website development. So the code review focused on ASP.NET/VB.NET code instead of HTML/CSS code. Conventional software coding standards and code review techniques can be adopted for the CRP.

In semester 2, 2005, 22 students (85%) were male and 4 (15%) were female. They came from seven different ethnic groups (Cambodian, Chinese, Indian, Japanese, New Zealander, Norwegian and Sri Lankan).

In semester 2, 2006, 14 students (78%) were male and 4 (22%) were female.

They came from six different ethnic groups (Chinese, Indian, Malaysian, New Zealander, Sri Lankan and Swiss).

In Semester 1, 2007, 9 students (70%) were male and 4 (30%) were female. They came from five different ethnic groups (Chinese, Indian, Iranian, Malaysian and New Zealander).

In Semester 1, 2007, 5 students (50%) were male and 5 (50%) were female. This is the only semester that there was not a male majority. They came from six different ethnic groups (Chinese, Dutch, Indian, Japanese, New Zealander and Thai).

Table 1. Students' age distribution

	Group 1 21-25	Group 2 26-30	Group 3 31-36	Group 4 50
2005	13	7	5	1
2006	10	4	3	1
2007 Sem1	7	1	0	4
2007 Sem2	6	0	2	2

The students' age distribution for the four semesters is shown in Table 1. From the above information we can see that the students in all the four semesters had the following similar features:

- young majority
- multiple cultural backgrounds.

The Initial Cycle

The initial problem, identified and analyzed by Li and Prasad (2005), was that the students were reluctant to apply the coding standards. As a possible solution, the CRP was designed and implemented in Semester 2, 2005. A survey, the lecturer's observations and an informal interview were used to monitor the CRP in this cycle. The main objectives of the CRP were (Li, 2006):

- to motivate students to learn coding standards
- to facilitate communication amongst the students.

So the CRP was not intended to be either a major assessment instrument or a collaboration learning tool. Therefore, the students were not trained in assessment skills or peer review skills. The students were expected to reinforce their coding standards knowledge and learn peer review skills from the process.

The main features of the CRP were (Li, 2006):

- face-to-face review
- the students selected their own reviewers
- the reviews contributed small weight to the course assessment (10% of Assignment 1 which in turn weighted 40% of the whole course)
- the reviews were used as a facilitating tool for coding standards
- the reviews were a part of the quality assurance for Assignment 1.

The CRP was carried out in the following sequence:

1. The students were given code guidelines which specified the coding standards required in the course.
2. The students were given a document which described the steps to be followed during the CRP.
3. The students were given a marking scheme to be used during the CRP.
4. Each student then chose two peers to conduct the CRP.

5. Each student submitted the reports from the two peer reviewers.

6. The CRP mark for each student was the average of the marks given by his/her two peers.

At the beginning of 2006, an electronic survey was sent to the 2005 class by email. Out of 23 students who participated in the CRP, 12 responded (a response rate of 52%). In this survey, questionnaires were used to obtain students' opinions and check the results of the CRP. There were seven rating scale questions and one selection question. Possible answers and corresponding scores for the rating scale questions are shown in Table 2 and the questions are shown in Table 3.

Table 2. Answers and scores for the rating scale questions

Answer	Scores
Strongly Disagree	0
Disagree	1
Neutral	2
Agree	3
Strongly Agree	4

Table 3. Survey questions

No.	Question
1	The CRP motivates you to think about more than coding standards
2	The CRP is helpful for your learning on coding standards
3	The CRP motivates you to follow coding standards
4	The CRP is helpful for you to improve your communication with your peers
5	The CRP is helpful for you to make more friends
6	The CRP is helpful for you to know more people with different cultural background
7	The CRP is helpful for you to learn and practise your critique skills
8	How many of your code reviewers were you not familiar with before this course?

The analysis of the 2005 results suggested that some of the students felt that the CRP assessment was unfair, because their peers were not well equipped with the necessary skills to assess their work. It also suggested that, in order to avoid the negative impact of "mutual admiration societies", the students should not choose the reviewers themselves (Li, 2006).

The Second Cycle

The following improvements were introduced in 2006:

- the two reviewers were assigned randomly by the lecturer
- the lecturer marked each student's work according to the same marking scheme as used by the students
- the CRP mark was calculated as the average of the lecturer's mark and the average marks given by his/her peers.

At the beginning of 2007, an electronic survey was sent to the 2006 class by email. Out of 12 students who participated in the CRP, 7 responded (a response rate of 58%). In addition to the survey, the lecturer's observations and the lecturer's informal conversations with the students were also used for this cycle. To check the results of the improvements introduced in 2006, the following additional question was included in the survey:

Do you prefer the reviewers assigned by the lecturer or the reviewers chosen by

yourself? (Assigned = 0; Neutral = 1 Chosen = 2)

A summary of the survey results for the initial cycle and the second cycle is shown in Table 4.

Table 4. Summary of the survey results

No.	Average 2005	STD 2005	Average 2006	STD 2006
1	2.83	0.83	3.00	1.00
2	2.67	1.15	2.86	1.07
3	2.75	1.06	3.14	1.07
4	2.67	1.37	3.43	0.53
5	2.50	1.17	2.57	0.53
6	2.42	1.08	3.14	0.69
7	2.67	1.23	3.00	1.15
8	0.83	0.72	1.43	0.79
9			1.14	1.07

For Questions 1 to 7, the average scores of the second cycle were consistently higher than those of the initial cycle. These suggested that the students in the second cycle liked the CRP more than the students in the initial cycle, which in turn suggested that the improvements introduced in this cycle were helpful. The average score for Question 8 was much higher in the second cycle than in the initial cycle, probably because the reviewers were assigned randomly. Question 9 had high standard deviation which indicates that the students' opinions were diverse on choosing reviewers.

The standard deviations for Questions 1, 2, 3 and 7 were still quite high, which indicates that there was a minority of the students who did not feel positive about the CRP. To investigate this further, the samples were divided into four groups: male students, female students, local-born students and overseas-born students. Scores did not vary greatly with age. Table 5 shows the group average scores for each question in the initial cycle. Table 6 shows the group average scores for each question in the second cycle.

Table 5. Group average scores of the initial cycle

No.	Male	Female	Local	Overseas
1	2.9	2.5	2.33	3
2	2.8	2	2.33	2.78
3	2.8	2.5	2.67	2.78
4	2.8	2	2.33	2.78
5	2.7	1.5	2	2.67
6	2.7	1	2	2.56
7	2.8	2	2.33	2.78
8	0.7	1.5	1.33	0.67
9	2.9	2.5	2.33	3

Table 6. Group average scores of the second cycle

No.	Male	Female	Local	Overseas
1	3.5	2.33	2	3.4
2	3	2.67	2	3.2
3	3.5	2.67	2	3.6
4	3.75	3	3	3.6
5	2.75	2.33	2	2.8
6	3	3.33	3	3.2
7	3.5	2.33	2	3.4
8	1.5	1.33	2	1.2
9	2	0	0	1.6

Tables 5 and 6 demonstrated that, for both cycles, the average scores of the male students were generally higher than those of the female students and the average scores from the overseas students were consistently higher than those of the local students. This suggested that in general male students and overseas students are happier with the CRP than female students and local students, respectively. This was consistent with the author's observations (Li, 2006). One possible interpretation is that the overseas students missed their families in their home country, so they needed more social life. On the other hand, the local students had enough social life in their homes and the local communities.

Another point worthy of notice is that the average scores for Question 5 (which was about making friends) were low for all groups in both cycles. On the other hand, Questions 3 (which was about motivation) and 4 (which was about communication) had reasonably high average scores in both cycles. The average score for Question 6 (which was about getting to know more people from other cultures) was much higher in the second cycle, probably because the lecturer assigned the groups randomly.

The average score for Question 7 (which was about critique) was 2.67 for the initial cycle and 3 for the second cycle. This suggested that the CRP did help in gaining critique skills, however, it was not a great help. In particular, the improvements introduced in the second cycle did not make much difference in this area. As critique skills should be an important learning outcome of the peer review process (Kern, Saraiva & dos Santos Pacheco, 2003), there is a big room for future improvement. The lack of the critique skills would have a negative impact on the quality of the CRP. As one student commented:

- How could a student give another student an accurate grade? They would give them all As.

In 2005 and 2006, it was observed that the CRP grade for a student was in general higher than his/her final grade (Li, 2006, 2007). One reason for this could be the students' lack of peer review skills and coding standards knowledge. Another reason could be the students' superficial engagement; they just quickly gave their peers a satisfactory mark and returned to their own work. These are consistent with some of the students' comments (Li, 2007). To address the issues identified in this cycle, more effective teaching strategies were required. Li (2007) proposed a number of the teaching strategies to help the students gain peer review related skills. These included:

- providing a training document
- using benchmark CRP samples
- letting the students do a self code review before their peer code review.

Li (2007) also proposed allowing the students to submit their code review one week later than their other documents, so the students would have enough time to focus on the code review. Using existing online tools for the CRP was also proposed.

The Third Cycle

The CRP in 2007, which was the third cycle, was carried out in much the same way as

the CRP in 2006 with one exception: the students were allowed to submit the CRP reports one week after they had submitted their assignments. The rest of the teaching strategies proposed by Li (2007) were not implemented in the third cycle. One of the reasons was that to implement these strategies, the students would have more workload. This would have required a redesign of the course assessment scheme which involved a formal procedure. Another reason was that this set of strategies focused on a subset of the skills that could be learned from the CRP.

As indicated in the last section, it was observed that the 2005 and 2006 students in general had given their peers much higher marks than their final grades. The same fact was observed in both semesters of 2007. To get more information about this fact and information about the students who did not participate in the CRP, two more observations were included. One observation was the difference between the lecturer's code review mark and the peers' average code review mark. This information would be helpful in determining the type and the amount of the training the students needed for the CRP. Another observation was the final grade for each student who did not participate. This information might be helpful to understand why these students were not interested in the CRP. Both observations were back-tracked to 2006.

Tables 7, 8 and 9 show all the relevant marks for each student who attempted Assignment 1 in 2006 and 2007. We can see that the average CRP mark given by the two peers for a student was generally higher than the CRP mark given by the lecturer for the same student. It was also generally higher than the same student's Assignment 1 mark and final mark.

Table 7. Summary of marks in 2006

Lecturer	Peers' Average	Difference	Assgn1 Mark	Final Mark
50	80	30	57.4	60
76	83	7	63.4	58
50			52.4	53
64	70	6	52	20
54	65	11	52	20
50	69.5	19.5	61.9	60
66	80	14	69.2	69
65	64.5	-0.5	55.9	57
81			76.4	81
83	99.36	16.36	81	83
56			56.7	60
50			43.4	50
90	87	-3	91.4	90
84	91	7	80.7	79
87	91	4	68.8	64
83	91	8	82.6	79
		Average = 9.95		

In 2006, of 16 students who attempted Assignment 1, 12 had participated in the CRP (a participation rate of 75%). Of the four students who did not participate in the CRP, three got C grades and one got an A grade. Table 7 shows that on average the CRP marks given by the students were 9.95 higher than the marks given by the lecturer.

In Semester 1 of 2007, of 12 students who attempted Assignment 1, 10 had participated the CRP (a participation rate of 83%). Of the two students who did not participate in the CRP, one got an E grade and another got an A grade. Table 8 shows that on average the CRP marks given by the students were 20.4 higher than the marks given by the lecturer.

Table 8. Summary of marks in 2007 Semester 1

Lecturer	Peers' Average	Difference	Assgn1 Mark	Final Mark
81	95.5	14.5	55	73
82	94.5	12.5	83	71
38	81	43	59	63
56			43	17
41	70	29	53	60
60	72	12	73	72
66	90	24	86	74
60	70.5	10.5	68	63
68	90.65	22.65	78	76
82			89	87
58	74	16	78	75
60	79.5	19.5	88	80
		Average = 20.4		

In Semester 2 of 2007, of 9 students who attempted the Assignment 1, 8 had participated in the CRP (a participation rate of 88.9%). The only student who did not participate in the CRP got an A grade. Table 9 shows that on average the CRP marks given by the students were 15.3 higher than the marks given by the teacher.

Table 9. Summary of marks in 2007 Semester 2

Lecturer	Student Average	Difference	Assgn1 Mark	Final Mark
76.7	87	10.5	87	80
65.3	74	8.7	63	57
47	64.5	17.5	48	53
91.3	0		87	80
50	77	27	44	64
79.3	95	15.7	84	81
61	84.5	23.5	75	73
78	87	9	86	78
71	81.5	105	74	50
		Average = 15.3		

To examine the whole process, participation information from the initial cycle was recalled. In 2005, of 25 students who attempted Assignment 1, 23 had participated in the CRP (a participation rate of 92%).

In summary, the average difference between the lecturer's mark and the students' mark is about 15%. This suggested that the peer assessment is far from accurate. The improvements introduced in the process did not make much difference in this area. The participation rate was not stable, varying between 75% and 92%, but there were always a small number of the students who do not participate in the CRP. Of the seven students who did not participate in 2006 and 2007, three got A grades overall and an A from the lecturer's code review, three got C grades overall and a C from the lecturer's code review, and one got an E grade overall and a C from the lecturer's code review. Considering the students who got an A final grade as high performing students and the students who had got a C final grade (or lower) as low performing students, this suggested that the high performing students had good knowledge about the coding standards and the low performing students had relatively poor knowledge about the coding standards. Some of the possible reasons for not participating in the CRP were identified by Li (2007) and the following are two additional possible reasons:

- the high performing students might not have much to learn from the CRP, so they are not interested in the CRP
- the low performing students might not have much skill/knowledge to contribute in the CRP.

Discussion on a Possible Framework

In this section, the findings from the above three cycles are highlighted and analysed further, and a possible framework is discussed. The CRP did well in achieving the two objectives proposed in the initial cycle (Li 2006, 2007). The improvements introduced in the second cycle did improve the CRP in terms of achieving the initial objectives. However, these objectives are too simple, and might limit the benefits the students can get from this process, and thus discourage the students from participating.

The data analysis of the second cycle suggested that there were diverse opinions about the CRP among different student groups, namely, male students were happier with the CRP than female students and overseas students were happier with the CRP than local students. The data analysis in the third cycle suggested that the students from low and high performing groups were less interested in the CRP for different reasons.

The data analysis of the second cycle suggested that the CRP did help students in gaining critique skills, however, it was not a great help. In particular, the improvements introduced in the second cycle did not make too much difference in this area. The data analysis of the third cycle suggested that the CRP should facilitate more learning to attract students' participation.

To address the issues identified in the process more effectively and to meet the students' needs better, the existing CRP needs to be refined to produce a framework. The CRP consists of two parts: the discussion part and the assessment part. The discussion part is a type of cooperative learning. If we expect the students learn more from the CRP, the discussion part needs to be more effective. Kern, Moore and Akillioglu (2007) summarised four essential elements for cooperative learning based on the existing theory:

1. There is evidence of group cohesiveness for accomplishing the task.
2. Individual group members take responsibility for individual efforts and contributions towards the team.
3. Use ways to improve the processes team members use to maximize their learning.
4. Promote one another's success, through a supportive, encouraging, and praising environment

The above four elements can be used as a starting point to discuss what is needed in the code review framework. The first element means that group members should work together to accomplish the task. The second element is that every member should make an effort to contribute. This requires that every member is equipped with enough knowledge and skills to participate. As identified in the third cycle data analysis, the low performing students do not meet this requirement, so training is necessary. The third element is that the process should be designed to maximize the learning. The techniques that can be used here include peer feedback and reflection. For our code review process, a reflection session can be included at the end of each review. The fourth element is to create a positive review environment. This can be achieved by carefully constructing the review structure and improving the students' skills. In the rest of this section, the code review framework is discussed from five aspects: training, structure, learning, format, and process.

Training is necessary for a more effective code review process. Peer review as a collaborative learning technique (Carlson, Berry, & Voltmer, 2005; McGourty, Dominick, & Reilly 1998) helps students to learn both course-related knowledge and collaborative skills (University of Michigan 2008). Collaborative skills include technique skills and critique skills. Critique skills should be an important learning

outcome of a peer review (Kern, Saraiva, & dos Santos Pacheco, 2003). Lack of critique skills would have a negative impact on the quality of the code review. The technique skills are subject-related and the course is about programming and software development, so the students should be trained on software code review techniques. The training should make the students understand how valuable code review is and what they can learn from the process.

The code review structure is about how to form the review team. Two different strategies had been used in the past three cycles. In the initial cycle, students chose their own reviewers. In the second cycle, two reviewers were randomly assigned by the lecturer. Chan et al. (2007) suggested that heterogeneous groups should be used rather than homogeneous groups, so the students can complement each other in their thinking styles, knowledge, roles and abilities. From the data analysis of the past three cycles, a number of student groups have been identified: male students, female students, local students, overseas students, high performing students and low performing students. The number of students in each group should be restricted when forming a code review team. It should be also noticed that this course usually attracts fewer female students than male students, and fewer local students than overseas students.

The code review learning is about what content we expect the students to learn from the code review. The students should learn more about the coding standards, and they should be asked to add any items identified during the code review, but not already in the given coding standards. Olds, McKenna and Pazos (2007) indicated that knowledge construction is social and so peer discussion is an effective technique to promote conceptual understanding. To make the code review more efficient and more attractive to the high performing students, the course content learning should be introduced into the code review as well. However, this content should be selected very carefully, so the students would not be distracted from the code review. For that reason, the content to be learnt should be relevant to the code review. Code review needs the reviewers to have an overview understanding of the code under the review - this is an opportunity for the students to have a better understanding about the code architecture. The course requires the students to understand the three tier architecture of a web application as well as the three layers of the code that reside on the web server. The code owner could be required to prepare the code architecture before the review, and the reviewers could verify the architecture by discussing it with the code owner.

The code review format is about how the code review is conducted. In general, we have online and offline options. For each of the options, we have asynchronous approaches and synchronous approaches. The online option needs to use electronic tools. Online synchronous code review requires the review team to stay online at the same time. A tool like Blackboard chat room (Blackboard, 2007) can be used for this purpose. The positive side of this approach is that synchronous discussion can happen and it could be anonymous for confidentiality purposes. The negative side of this approach is that all the members of the review team must have access to the Internet and the tool at the same time, thus it is not flexible. Online asynchronous code review means the code owner submits the code using an electronic tool such as email, Turnitin (Turnitin 2007) or a forum, the reviewers then review the code and give their feedback to the code owner later. The positive side of this approach is that the time is flexible, so the reviewers will have sufficient time to refine their comments, thus their comments should be more appropriate. The negative side of this approach is that all the members of the review team must have access to the tool or the discussion will not happen.

Offline synchronous code review means that the code owner and the reviewers will have a face-to-face meeting to discuss the code. This is the approach used in our past three cycles. The positive side is that the students can learn effectively from the discussion; the negative side is that every team member has to come to a chosen venue at the same time, thus it is not flexible. Offline asynchronous code review means that the code owner submits the code, then the reviewers review the code and give the owner feedback. This approach has been used for the lecturer's code review in the second and the third cycles. The positive side is that the reviewers have

sufficient time to refine their comments, thus their comments should be more appropriate; the negative side is that there is no discussion.

The code review process defines the steps to be followed in the code review. To promote effective learning, the same model used in our second and third cycle (the offline synchronous code review approach) will continue to be used. In addition to that, a reflection session will be added after each code review. Each code owner will fill a form to comment on their peers' critique skills and technique skills, to identify the issues not included in the existing coding standards and to revise the existing coding standards.

A Proposed Framework

According to the discussion in the last section, a framework for future code review is proposed. The framework consists of five components as discussed in the last section: training, structure, learning, format, and process. The code review training component consists of a number of documents. Some of them have been proposed by Li (2007):

- A coding standard training document which answers frequently asked questions about coding standards.
- Four benchmark code review samples for poor, good, fine and excellent work respectively.
- A training document which covers the rationale of the code review, the learning outcomes of the code review and the process of the code review.
- A peer review skill document which covers the correct attitude for the code owners, the correct attitude for the code reviewers, and the characteristics of reflective feedback.
- A technique skill document which covers software code review techniques. Examples of these include control flow analysis, dataflow analysis and user interface analysis (Meier et al., 2005).

The code review structure defines the code review team members. Each review team will consist of three members: a code owner and two peer reviewers. The peer reviewers will be assigned by the lecturer (as much as possible) according to the following principles:

- each team will have at least one female student
- each team will have at least one local student
- each team will have a balance of knowledge and skills.

The code review learning component contains a training document for the course content to be learned in the code review process. An example of this is a sample document about the three layer code architecture. The code review format is the offline synchronous code review approach.

The code review process is a revised version of the one described by Li (2006). The code review will be carried out in the following sequence:

1. The students will be given code guidelines which specify the coding standards required in the course.
2. The students will be given a marking scheme to be used during the code review.
3. The students will be given all the training documents stated in the code review training component.
4. The students will do a self code review.
5. The students will be given a document which describes the steps to be followed in the team code review.
6. Each student will be assigned two peer reviewers by the lecturer according to the principles stated in the code review structure component, and then a review team will be formed for each of the students.

7. The students will be given a reflection form to fill after their own team code review.
8. Each student will provide architecture information for their code to be reviewed.
9. The team code review will be conducted for each of the students.
10. Each student will submit one self code review report, two peer review reports and one code review reflection form.
11. Each student will get credit for submitting their self code review report and reflection report plus the average of the marks given by his/her two peer reviewers.

Summary and Future Plans

This paper reviews the past three cycles of the CRP and tries to get deeper understanding about the following issues:

- The CRP assessment was far from accurate.
- The CRP did not maximize learning of the course content.
- The male, female, local and overseas student groups have diverse opinions about the CRP. However, there was no sign of the diverse opinions from age groups.
- The students who did not participate in the CRP were either high performing students or low performing students. To address the above issues more effectively and meet the students' needs better, the existing CRP has been refined to produce a framework consisting of five components. The proposed framework will increase the weight of the code review in the course assessment. This will increase the workload for both of the lecturer and the students. To address this issue, the course assessment scheme needs to be redesigned, for example, introducing group work or removing the final exam. Initially, the framework should be implemented in the same course. If this implementation is successful, there is a potential to revise the framework and introduce it into similar courses at Levels 5, 6 and 8. The data samples used in this research were too small to get a complete understanding of the students' opinions. This might have a negative impact on the proposed code review framework.

References

- Blackboard. (2007). *The Blackboard Content System*. Retrieved March 11, 2007, from http://www.blackboard.com/products/Academic_Suite/Content_System/index.
- Bogue, R. (2006). *Effective code reviews without the pain*. Retrieved March 11, 2007, from <http://www.developer.com/java/other/print.php/3579756>
- Carlson, P., Berry, F., & Voltmer, D. (2005). Incorporating student peer-review into an introduction to engineering design course. *Proceedings of 35th ASEE/IEEE Frontiers in Education Conference* (pp. F2C20- F2C25).
- Chan, T., Cheng, Y., Wu, Y., Jong, B., & Lin, T. (2007). Applying learning achievement and thinking styles to cooperative learning grouping. *Proceedings of 37th ASEE/IEEE Frontiers in Education Conference* (pp. T1C9-T1C13).
- Dimon, G. (2006). *Code reviews: Write better code overnight*. Retrieved March 11, 2007, from http://www.digital-web.com/articles/code_reviews_write_better_code_overnight/.
- Kern, V. M, Saraiva, L. M., & dos Santos Pacheco, R.C. (2003). Peer review in education: Promoting collaboration, written expression, critical thinking, and professional responsibility. *Education and Information Technologies*, 37-46.
- Kern, A., Moore, T., & Akillioglu, F. (2007). Cooperative learning: Developing an observation instrument for student interactions. *Proceedings of 37th ASEE/IEEE Frontiers in Education Conference* (pp. T1D1-T1D6).
- Li, X., & Prasad, C. (2005). Effectively teaching coding standards in programming. *Proceedings of SIGITE Conference* (pp. 239-244).
- Li, X. (2006). Using peer review to assess coding standards – A case study. *Proceedings of 36th ASEE/IEEE Frontiers in Education Conference* (pp. T2D9-T2D14).
- Li, X. (2007). Incorporating a code review process into the assessment. In S. Mann, & N. Bridgeman (Eds.) *Proceedings of 20th Annual Conference of the National*

- Advisory Committee on Computing Qualifications*, (pp. 125-131). Nelson: NACCQ.
- McGourty, J., Dominick, P., & Reilly, R. (1998). Incorporating student peer review and feedback into the assessment process. *Proceedings of Frontiers in Education Conference, Session T1A/em*>.
- Meier, J., Mackman, A., Wastell, B., Bansode, P., Taylor, J., & Araujo, R. (2005). *How to: Perform a security code review for managed code (baseline activity)*. Retrieved February 18, 2008, from [http://msdn2.microsoft.com/en-us/library/ms998364\(printer\).aspx](http://msdn2.microsoft.com/en-us/library/ms998364(printer).aspx) .
- O'Brien, R. (2001). *An overview of the methodological approach of action research*. Retrieved May 4, 2007, from <http://www.web.ca/~robrien/papers/arfinal.html> .
- O'Connor, T. (2007). *Lecture notes, Survey research design*. Retrieved March 11, 2007, from <http://faculty.ncwc.edu/toconnor/308/308lect07.htm> .
- Olds, S., McKenna, A., & Pazos, P. (2007). Work in progress - Promoting conceptual understanding through effective peer discussions in large classes. *Proceedings of 37th ASEE/IEEE Frontiers in Education Conference (pp. T1D7-T1D8)*.
- Parasoft. (2005). *Understanding the workflow in a coding standards implementation*. Retrieved March 4, 2005, from <http://www.parasoft.com/jsp/pr/runtimes.jsp?runtimeId=1169>
- Parasoft. (2007). *Source code review process*. Retrieved March 11, 2007, from http://www.parasoft.com/jsp/aep/aep_practices.jsp?practice=CodeReview
- Pfeiffer, J. (2005). *Coding standards in 400- and 500-level classes*. Retrieved July 1, 2005, from. <http://www.cs.nmsu.edu/~pfeiffer/classes/general/s05/coding.html>.
- Savenye, W. C., & Robinson, R. S. (2003). Qualitative research issues and methods: An introduction for educational technologists. In D. H. Jonassen (Ed.), *Handbook of research on educational communications and technology* (2nd Ed., pp. 1045-1071). Retrieved March 11, 2007, from <http://www.aect.org/edtech/39.pdf>
- Trahasch, S. (2004). From peer assessment towards collaborative learning. *Proceedings of 34th ASEE/IEEE Frontiers in Education Conference (pp. F3F16-F3F20)*.
- Turnitin. (2007). *Peer review: Turnitin's collaborative learning system*. Retrieved March 11, 2007, from https://turnitin.com/static/pdf/datasheet_preview.pdf
- University of Michigan. (2008). *Teamwork & group work*. Retrieved February 17, 2008, from http://www.engin.umich.edu/teaching/assess_and_improve/handbook/direct/teamwork.html
- Web Center for Social Research Method. (2007). *Trend studies*. Retrieved March 11, 2007, from <http://www.socialresearchmethods.net/tutorial/Cho2/trend.html> .
- Weinreich, N. (1996). *Integrating quantitative and qualitative methods in social marketing research*. Retrieved March 11, 2007, from <http://www.social-marketing.com/research.html>.
- Xiaosong Li Unitec New Zealand xli@unitec.ac.nz Donald Joyce Unitec New Zealand djoyce@unitec.ac.nz Towards a Framework for a Code Review Process

Copyright © 2008 Li, X.

Journal of Applied Computing and Information Technology (JACIT): ISSN 2230-4398

(Incorporating the Bulletin of Applied Computing and Information Technology, NACCQ: ISSN 1176-4120 and Journal of Applied Computing and Information Technology, NACCQ: ISSN 1174-0175)

Copyright ©2008 CITRENZ.

The author(s) assign to CITRENZ 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 author(s) also grant a non-exclusive licence to CITRENZ to publish this document in full on the World Wide Web (prime sites and mirrors) and in printed form within the Journal of Applied Computing and Information Technology. Authors retain their individual intellectual property rights.

Donald Joyce (Editor).

