Assessment of programming language learning based on peer code review model: Implementation and experience report

Yanqing Wang,*, Hang Li, Yuqiang Feng, Yu Jiang, Ying Liu

School of Management, Harbin Institute of Technology, Harbin 150001, China
Department of Information Systems, College of Business Administration, California State University Long Beach, CA 90840, USA

Abstract

The traditional assessment approach, in which one single written examination counts toward a student’s total score, no longer meets new demands of programming language education. Based on a peer code review process model, we developed a novel online assessment system called EduPCR and used a novel approach to assess the learning of computer programming languages. Using this approach, students peer review programs written by other students, share ideas and make suggestions to achieve an objective of collaborative and interactive learning. Teachers assess and give scores to students based on their performance in writing, reviewing and revising programs and their abidance to a peer code review process. After using this approach in two courses in two consecutive semesters, we observed significant improvements of student learning in various aspects. We also conducted two questionnaire surveys and two interviews. The survey data and the interview report indicated that this assessment approach demonstrates high practical values in assessing student learning outcomes in programming languages. Additionally, this approach leads to several interesting research topics for future research in this field.

1. Introduction

Among various aspects of education, the assessment of learning outcomes is critical in ensuring good pedagogical quality and has always been difficult. Because computer programming is problem-solving oriented and very practical, the assessment of programming learning performance is challenging. The traditional assessment is not easy to be adapted to various new developments in computer programming education. New teaching approaches such as collaborative learning, project-based learning (PBL), e-Learning and m-Learning are endeavors of exploring new ways to boost learning outcomes.

Code review is a quality assurance mechanism at source code level. It was originated from software engineering and has been a common programming practice for many years. Fagan (1976) first introduced the concept of code inspection and code review. Afterward, the feasibility and high efficiency of code review has been studied and testified by many researchers (Mäntylä & Lassenius, 2009; Takagi et al., 1995). There are many widely-used commercial code review software products such as Code Collaborator from SmartBear Company in United States and Crucible from Atlassian Company in Australia. The research of code review includes processes and approaches (Ballantyne, Hughes, & Mylonas, 2002; Fallows & Chandramohan, 2001; Jacob & Pillai, 2003), behavior enhancement and quality assurance (Cunha & Greathead, 2007; Maldonado et al., 2006) and supporting tools (Belli & Crisan, 1996; Oh & Choi, 2005; Remillard, 2005; Silva & Moreira, 2003).

Observing the successful practices of code review in software industry, many computer science educators have shown their interests in introducing peer code review (PCR) processes to their courses (Gehringer, Chinn, Pérez-Quiñones, & Ardis, 2005; Sithiworachart & Joy, 2004; Trytten, 2005). A number of good learning outcomes have been reported (Li, 2006, 2007; Turner & Pérez-Quiñones, 2009). Turner (2009) described positives effects of code review in teaching freshmen concepts of object oriented programming. Li (2006) assessed coding quality when students used code review in programming assignments. Additionally, Li (2007) applied code review process in course...
assessment and analyzed the learning outcomes. However, the above research seldom developed and applied assessment software system and failed to investigate quality assurance thoroughly.

Our research on PCR dated back to 2006. The publications in the past five years focused on process improvement (Wang, Li, Collins, & Liu, 2008; Wang, Su, Hu, & Wang, 2007), quality assurance (Wang, Xu, Su, & Liu, 2008; Wang, Yang, Liu, & Collins, 2007; Wang, Zhang, Yu, & Huang, 2008) and learning outcomes (Wang, Li, Sun, Jiang, & Yu, 2011). We designed and implemented a PCR system and applied it in the assessment of two programming language courses in 2010. The PCR model was used as a formal assessment approach in those courses. The assessment approach covered the following areas: the competence in writing program, the carefulness and responsibility of learning attitude, the degree of compliance with coding standards, and the capability of abiding by schedules.

This paper is structured as follows. Section 2 introduces the PCR process model and an assessment software system that was developed for programming language education. Section 3 describes different aspects of the proposed assessment approach: assignment and grading, approach features and quality assurance strategies. Section 4 presents analytic results of two case studies. Section 5 gives the experiment report that includes the approach’s impacts on students and remaining problems. Conclusions and future work are discussed in Section 6.

2. PCR model and EduPCR

2.1. PCR model

The PCR model consists of five roles described in Table 1 and three documents described in Table 2. Students play the roles of author, reviewer and reviser in different steps of each assignment to write, review and revise programs. A teacher role is responsible for giving assignments and setting schedules. Only a teacher role or a teacher assistant role can inspect and grade student submission. The assessment process begins with a new assignment from a teacher and ends with score check by students. The review process is specified by an activity diagram depicted in Fig. 1.

2.2. EduPCR system

To meet the requirements of programming language learning assessment, we designed and developed a peer code review software system called EduPCR. The software system has been developed in two versions. The version 1.0 system supports only a single course with limited functions and poor expandability. The version 2.0 system has significant improvements in many aspects. It uses best practices of code review and greatly simplifies the management of code review process.

Because this system is dedicated to learning assessment in programming education and its developers and users are mainly students, we used many open source techniques. The development framework is based on JavaEE that consists of Struts 2, Spring and Hibernate. The Web server uses Apache Tomcat 7 and the database server is MySQL 5. The development environment uses MyEclipse 8.5.

Fig. 2 gives an example of system user interface. After login, a student sees assignments he/she is working on. Fig. 2 shows that student 101000206 has entered an activity in an Object Oriented Programming in Java course. The task “Matching TelNo” has been completed and is to be graded by a teacher. For the task of “Create a voice script database”, this student should review code written by another student. The task “Matching braces” is completed and the student can check the grade of this task.

Fig. 3 shows a code review task. In this web page a student can browse program source code written by another student, add line-level comments on the source code through mouse positioning and clicking, and give comprehensive comments and evaluation (peer evaluation among students for teachers’ reference) in the text fields below the whole program.

3. Assessment approach

3.1. Assignments and grading

For each course, after training students on peer code review, the course teacher creates a certain number of assignments (10 assignments in both cases of this research) with medium difficulty and schedules them at specific intervals (one week on average in this research). The due time of each assignment should be carefully managed by teachers to assure learning quality and prevent students from being too tired and nervous.

The scores of each assignment are classified into process scores and quality scores as specified in Table 3. To be more feasible, the version 2.0 system enables the teacher in charge of a course to manage the score distribution of each assignment according to its difficulty and students’ programming capability. As for student, if one wants to get a higher score, he/she should meet the requirements listed in Table 4 and avoid the typical cases of losing scores listed in Table 5.

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>The roles in the model.</td>
</tr>
<tr>
<td>Author</td>
</tr>
<tr>
<td>Reviewer</td>
</tr>
<tr>
<td>Reviser</td>
</tr>
<tr>
<td>Teacher</td>
</tr>
<tr>
<td>TA</td>
</tr>
</tbody>
</table>
3.2. Features of the assessment approach

The proposed assessment approach has the following desired features:

(1) **Rigorous process restraints.** The software system helps to realize rigorous process management. To prevent students from skipping steps of an assignment and to minimize influence on a student whose code is not reviewed by another student on time, each student is required to finish each assignment in a fixed order – writing and submitting manuscript code, reviewing designated code written by another student, revising reviewed code and re-submitting revision code. When multiple assignments are in process, the task order restraint is effective only within an individual assignment. Additionally, one student’s failure to submit a task on time could not affect the evaluation of another student. There are three scenarios: i) if a student fails to submit manuscript code before its deadline, the related reviewer designation list will be compacted and this student will be removed from the review task; ii) if a student fails to submit review comments before its deadline, the system will automatically “finish” the review with comments like “Because the reviewer did not complete the code review, please revise your code according to your own knowledge and understanding”; iii) if a student fails to submit revision code before the last deadline, there will be no influence on other students.

(2) **Open assessment environment.** The assessment is neither confined to a specific classroom nor at a certain short period of time so that students can work on their paces in places they choose. Additionally, they can refer to rich web resources, discuss with their classmates and even ask teachers for help using the assessment software system. Nonetheless, students are required to work independently on their code and plagiarism is strictly prohibited.

(3) **Real-time assessment.** Following the effectiveness principle of the reinforcement theory by Skinner (1938), a major feature of the assessment approach is its real-time assessment. Upon finishing an assignment, a student can query all cumulative comments, see real-time assessment results submitted by teacher and check his/her learning progress online anytime.

(4) **Instant message update.** The system is equipped with a short message gateway (GSM Modem) to send text messages to students’ cell phones instantly. Students receive reminders of events such as new assignment, code review and code revision when tasks are available in the system.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>The documents in the model.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript code</td>
<td>The source code of a program written by an author (a program passed steps of compiling, building and testing)</td>
</tr>
<tr>
<td>Review comments</td>
<td>The ideas, suggestions, and criticism that a reviewer gives to an author, mainly including coding standards, fatal defects, design logic, redundant code and non-functional requirements such as simplicity and performance.</td>
</tr>
<tr>
<td>Revision code</td>
<td>A new edition of program submitted by a reviser based on the review comments</td>
</tr>
</tbody>
</table>

**Fig. 1.** Activity diagram of EduPCR system (version 2.0).
Integrated learning approach. This assessment approach is based on a number of small projects and requires many interactions among students. The system is accessed through the Internet and takes advantage of cell phone for notifications. Therefore, this assessment approach integrates diverse learning methods such as project-based learning (PBL), collaborative learning, e-learning and m-learning.

3.3. Quality assurance

The proposed assessment approach has a number of quality assurance strategies that include double-blind review, designation of reviewer, and peer evaluation.
(1) **Double-blind review.** To ensure the review equality and exclude personal relationship factors, the system uses two methods to accomplish double-blind review. Designating reviewer randomly means that, when designating reviewers, the system generates several designation rings randomly thus any student does not know who will review his/her program. When a reviewer reads a program through a web browser, the author name is masked as “anonymous” by the system. When a teacher inspects and grades a program, the author name is also hidden to avoid “marking by impression”.

(2) **Designation of reviewer.** The strategy of reviewer designation in PCR process has drawn attention of some researchers such as Turner (2009). Our version 1.0 system implements a fixed designation strategy that uses a mutual review by two students or a ring review by three students. The fixed designation strategy may introduce problems such as using substitute in writing code, conducting review or revising codes (Wang, Li, et al., 2011; Wang, Su, Ma, Wang, & Wang, 2011). Consequently, EduPCR 2.0 uses a random designation strategy as the strategy of reviewer designation. The designation algorithm has two major steps: i) randomly dividing the total \( N \) students of one class into \( m \) groups with at least 3 students in each group; ii) building a ring for each group such that a student reviews code of another student outside the system.

(3) **Peer evaluation.** To help learning assessment, the system requires a reviewer to comment and evaluate the source code written by another student. An author needs to respond to the comments made by the reviewer. Though review comments and peer evaluation are not “real” scores of a student, they are used as teacher’s references in student assessment.

### 4. Implementation and experiment report

#### 4.1. Case description

In 2010, EduPCR was applied in a pedagogical study of two undergraduate courses in Department of Management Science and Engineering at Harbin Institute of Technology.

**Case 1:** In the spring semester of 2010, the version 1.0 system was used in the assessment of a course titled *Object Oriented Programming in Java* for freshmen in Class 0910101 majoring in Information Management and Systems. There were 31 students in this course. The assessment had 10 programs whose inspection and evaluation were all completed by the teacher of the course.

**Case 2:** In the autumn semester of 2010, the version 2.0 system was used in the assessment of a course titled *C Programming* for freshmen in classes 1010001-3 majoring in Management Science and Engineering. There were a total of 87 students in three classes. The assessment also had 10 programs whose inspection and evaluation were completed by one teacher and two teacher assistants. In Case 2, the software system and the assessment approach were refined on the basis of Case 1 and several problems found in Case 1 study were resolved.

#### 4.2. Questionnaires and analytic results

At the end of each case study, the PCR research team conducted a questionnaire survey and an interview to evaluate the system and the assessment approach. The two surveys have similar content that includes a set of multiple-choice questions and a set of short answer questions. Questions were distributed to all students in both cases and there were 30 valid responses in case 1 and 79 valid responses in case 2.

The analytic results of the questionnaire can be classified into the following categories: degree of student satisfaction, rationality of double-blind review, acceptance of real-time assessment, attitude of giving and accepting criticism, functions of short message gateway, rationality of online peer evaluation, rationality of the grading scheme, and necessity of presenting error types in the review comments.

(1) **Degree of student satisfaction.** The students were generally satisfied with the PCR assessment approach as depicted in Fig. 4 and Fig. 5. Compared with traditional assessment approaches without PCR, about 80% (“like moderately” plus “like very much”) of the students considered that the PCR assessment approach is superior in several aspects. Through the interview, we gathered and understood the following students’ opinions on the PCR assessment approach:

<table>
<thead>
<tr>
<th>Every student</th>
<th>Completing each step of one assignment on time (as early as possible); using proper name, format and version in every submitted file; ensuring fair peer evaluation and constructive comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>The author</td>
<td>Carefully writing the manuscript code to make it correct and terse; following coding standards; avoiding errors</td>
</tr>
<tr>
<td>The reviewer</td>
<td>Responsibly pointing out as many problems (such as coding standards, mistakes and improvements) as possible in a program</td>
</tr>
<tr>
<td>The reviser</td>
<td>Carefully making revisions according to review comments (revisor can reserve different viewpoints to reviewer’s comments)</td>
</tr>
<tr>
<td>Every student</td>
<td>Completing an assignment independently rather than copying others’ codes; not writing, reviewing, and revising for other student outside the system</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th><strong>Table 3</strong></th>
</tr>
</thead>
</table>

Score distribution of each assignment.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Content</th>
<th>Full mark</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process score (45)</td>
<td>Writing manuscript code</td>
<td>15</td>
<td>Submitting on time</td>
</tr>
<tr>
<td>Peer reviewing</td>
<td></td>
<td>15</td>
<td>Submitting on time</td>
</tr>
<tr>
<td>Revising program</td>
<td></td>
<td>15</td>
<td>Submitting on time</td>
</tr>
<tr>
<td>Quality score (55)</td>
<td>TA inspects and grades the work review by reviewer</td>
<td>20</td>
<td>Review comments should be helpful and peer evaluation is fair</td>
</tr>
<tr>
<td></td>
<td>TA inspects and grades the manuscript code</td>
<td>20</td>
<td>Program should work and meet the assignment requirements</td>
</tr>
<tr>
<td></td>
<td>TA inspects and grades the revision code</td>
<td>15</td>
<td>Program should be revised according correct review comments</td>
</tr>
</tbody>
</table>
Typical cases of losing points.

<table>
<thead>
<tr>
<th>Case</th>
<th>Points lost</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failing to submit a program on time</td>
<td>100 points are lost</td>
<td>It is regarded as that a student abandons this assignment</td>
</tr>
<tr>
<td>Failing to submit a revised program on time</td>
<td>65 points are lost</td>
<td>30 (15 + 15) process points + 35 (20 + 15) quality points</td>
</tr>
<tr>
<td>Copying program code from others</td>
<td>30 points are lost</td>
<td>15 process points + 15 quality points</td>
</tr>
<tr>
<td>Failing to review a program on time</td>
<td>Conspired plagiarism, 80 points are lost for all students involved</td>
<td>Prohibited conduct</td>
</tr>
<tr>
<td>Plagiarism without permission, 80 points are lost</td>
<td></td>
<td>Possibly secret copy in review step</td>
</tr>
</tbody>
</table>

- PCR is a novel and interesting learning approach that can stimulate their interests in learning and enhance their awareness of active learning;
- Compared with traditional assessment processes, this process helps to improve the actual programming ability;
- This approach not only adds more fairness to assessment but also helps to achieve learning objectives more efficiently.

(2) Rationality of double-blind review. The survey results of Case 2 (Fig. 6) indicated that 72% of the students considered the double-blind review a rational process (this question was not in Case 1). They held the view that this mechanism could eliminate the factors of personal emotion. The PCR could produce a more objective attitude of making suggestions and accepting criticism and encourage students to share their opinions straightly. Interestingly, 50% of the students in Case 1 and 30% of the students in Case 2 wanted the system to display the reviewers' name thus they could clarify some confusing comments made by the reviewers. Additionally, a few students considered that publicizing author names could make authors more careful about their coding since they feared “losing face” when poorly written programs were reviewed by other students.

(3) Acceptance of real-time assessment. Most students applauded real-time assessment (See Fig. 7 and Fig. 8). When an assignment had been completed by all students or its last deadline had passed, the teacher would begin to assess students' work as soon as possible. After the teacher had submitted assessment results online, students could get real-time information of the assignment. As a result, students showed positive learning attitude and a sense of urgency in working on their assignments.

(4) Attitude of giving and accepting criticism. 94% of the students were pleased to or willing to accept critical comments (Fig. 9). Even when the comments on their code were wrong, most students could rationally handle them by giving validating response or asking helps from teachers (Fig. 10). Most students were able to present their criticism to authors without reservation (Fig. 11). Consequently, most students benefited from the double-blind review process because it helped them give and accept criticism in an open and supportive way.

(5) Functions of short message gateway. In Case 1, we observed that many students, especially active learners, went to a computer lab several times a day to check the availability of a new assignment or status updates of existing assignments. This not only brought students unnecessary worries and wasted their time but also lower the efficiency of the PCR learning process. In Case 2, the EduPCR 2.0 implemented a short message gateway that sent real-time notification to students' cell phones. The short message gateway played an important role in the PCR process. In order to test the value of the real-time notification, the short message gateway was closed in the sixth program assignment. The questionnaire results (Fig. 12) showed that after the short message gateway was closed, 68% of the students were affected by the “malfuction” of the short message notification.

(6) Rationality of online peer evaluation. The survey results of this item in the two cases have mediocre satisfactory (Fig. 13 and Fig. 14). One reason is that peer evaluations on student assignments were generally high. The high evaluations were attributed to the simple assignments of the introductory programming language courses. Another reason is that some students were confused by evaluations that were not described in details.

(7) Rationality of the grading scheme. The grading scheme plays an important role in assessment quality. In Case 1, students did not like the distribution of process scores (60%) and quality scores (40%) because it failed to offer students incentives to improve code quality. As a result, the grading scheme was changed in Case 2 that the process scores accounted for 45% and the quality scores accounted for 55%. In Case 2, students were more satisfied with the changed grading scheme than that in Case 1. Additionally, students proposed some suggestions to improve the grading scheme. For example, it helps to describe quality scores in details and give higher score to authors who write good code comments in their programs.
Necessity of presenting error types in the review comments. In Case 1, the comments in the PCR process were presented with error types including coding standard violation, logic mistakes, and unproved algorithm. In responses to the questionnaires, students pointed out that the error types were incomplete and more types should be added. Similarly, the responses to questionnaires in Case 2 showed that up to 87% of students hoped to increase error types to facilitate more accurate reviews and better revision.

5. Experiment report

5.1. Impacts on students

The PCR assessment process has been used in two programming courses and the two questionnaire surveys found some positive impacts in student learning in the following areas:

(1) Student programming skills. Compared with traditional assessments that concentrate on theoretical study, the PCR approach puts a high priority on practical programming competence. In the coding stage of an assignment, students are required to complete many hands-on practices such as defining variables, selecting algorithm, revoking function, compiling and debugging. Students need sufficient knowledge of coding concepts and practice a lot to meet coding requirements. In addition, student programming competence was improved significantly in the code review stage. As shown in the responses of questionnaires and interviews, students agreed that this process improves their programming skills from the following aspects:

- A student is able to enhance his/her programming skills and learn different programming techniques from reviewing programs written by other students;
- A student could gain a profound understanding when he/she tries to understand programs written by other students;
- The constructive comments from reviewers encourage a student to write a better program in the revising step;
- Being a reviewer gives a student a sense of responsibility that prompts active learning.

(2) Collaborative learning competence. The PCR-based assessment is an open process that gives students a lot of flexibility in completing their assignments. It was observed that most students were willing to openly discuss programming problems, especially in the review phase. The collaborative nature of the PCR assessment process made students feel connected to their classmates: they should carefully write their own programs and responsibly review programs written by other students because it is collaborative and mutually beneficial process. The carefulness and responsibility of each individual affected herself/himself as well as others who enroll in the same class.

(3) Compliance with coding standards. It is of vitally important for students to comply with well-accepted coding standards in learning a programming language from the very beginning. The code review process enforces coding standards in four programming areas: identifier names, layouts (whitespace, indentation, blank line, etc.), code comments, and control statements (if, for, while, etc.). Students learned many

Fig. 5. Degree of students’ satisfaction (Case 2).

Fig. 6. Rationality of double-blind review (Case 2).
incorrect coding styles from review comments and were able to find similar mistakes when they played the reviewer role. At the end of a course, students were found to improve a lot in complying with coding standards and the average number of violations of coding standards in a program was reduced from more than 10 at the beginning to almost zero at the end of the course.

(4) Time management capability. Rigorous process control helps to build students’ time management capability. Teachers set three deadlines for each assignment in the assessment system. Activities such as uploading manuscript code, submitting review comments and uploading revision code should be completed by student on time. If a student fails to complete a certain step on time, the activity web page will display “overdue” and the system will automatically take some points off from the step. The system data showed that almost all students submitted 10 assignments on time. At the semester end, 88% of the students reported that they had developed solid time management skills.

(5) Competence of giving and accepting criticism. Students should be able to give and accept criticism no matter whether they will work in academia or industry (Silva & Moreira, 2003). They are not only required to complete tasks independently but also have to rapidly find and correct mistakes in a team environment. Engineering education should teach students how to give unbiased and constructive criticism effectively. The code review step asks students to be good at discovering and pointing out problems in other students’ programs without reservation and at the same time accept criticism from others with grace and appreciation. Responses of the questionnaires indicated that most students could smoothly perform the review task. After the repetition of the same process of “learning–giving criticism–accepting criticism–relearning”, most students had achieved the objective of giving and accepting criticism kindly and effectively.

5.2. Remaining problems

We found some shortcomings of the assessment approach from student feedbacks and through our use of the PCR software system.

(1) Heavy workload for teachers. Our software system did not use an automatic assessment technique in the assessment process yet. A teacher or a teacher assistant needs to manually grade all three steps in each assignment. In Case 2, there were 87 students and 10 program assignments. Each program had two versions: the original submission and the revision (based on review comments). Consequently it was very time consuming to manually grade a total of 1740 programs. Moreover, the quality and consistency of the manual grading left lots of room for improvement.

(2) Reviewer designation issues. The PCR process uses a design of “random reviewer designation + double-blind review” that has brought good results. Nonetheless, due to the possible big variance of programming skills among students, the code review process did not reach its maximum benefit because of the randomness in reviewer designation. In other words, the current random designation strategy does not utilize any information of student programming skills.

(3) Plagiarism. Although assessment rules had been passed out to all students at the beginning of a class, there were several cases of plagiarism and some students could not finish assignments independently. Plagiarism severely eroded the credibility of the PCR assessment and decreased the learning outcomes.

![Fig. 7. Acceptance of real-time assessment (Case 1).](image1)

![Fig. 8. Acceptance of real-time assessment (Case 2).](image2)
Fig. 9. The initial reaction of students to criticism (Case 2).

Fig. 10. Attitude of students to possibly wrong criticism (Case 2).

Fig. 11. Thoughts of students in giving criticism to classmates (Case 2).

Fig. 12. The degree of impact on students during the close of the short message gateway (Case 2).
The phenomenon of the "last minute crunch". Some students hurried to submit their work just before deadline and their review comments could be too simple or make no sense. Our investigation showed that in some cases those students had not formed a good habit of time management. In other cases, some students feared that the sooner they finish an assignment, the sooner a teacher will give the next one.

6. Conclusions and future work

The EduPCR system and its PCR-based assessment process have significantly improved student learning outcomes in many areas including programming skills, collaborative learning, compliance with coding standards, time management, and giving and accepting criticism. Responses to questionnaires of the two cases showed that student satisfaction rate with this assessment system exceeded 80%. The PCR-based assessment could be used as an efficient assessment approach in programming language education.

Nonetheless, the assessment system has some issues in its design and implementation. Due to challenges in assessment of programming language education as well as issues in student learning motivation and learning behavior, there are some interesting research topics to be investigated in future research. We want to focus on the quality assurance in an assessment process and the incentive mechanism in student learning. The future research in quality assurance may work on the following topics:

1. Automatic assessment technique and game theory model. The heavy teacher workload is a big concern in PCR-based assessment process. A possible solution is to use automatic assessment techniques. When a sampling inspection approach is adopted, the unsampled students should be given fair grades using an automatic assessment technique. Some successful automatic assessment tools such as AutoLEP (Wang, Su et al., 2011), Web-CAT (Edwards & Perez-Quinones, 2008) and BOSS (Joy, Griffiths, & Boyatt, 2005) can be used to grade student program assignments automatically. Additionally, the assessment process can be improved by using a game theory model that includes awarding and punishing strategy. By properly adjusting the parameters of award and punishment in the review process, we expect that students will have more incentives to write source code carefully and review code responsibly.

2. Ranking-based reviewer designation strategy. By ranking students into different categories (high, medium, low, etc.) based on their programming capability we can apply two designation strategies. Equal-rank designation strategy designates students of equal rank in the same review ring. Stepwise designation strategy means that higher-rank students review programs written by lower-rank students and a teacher reviews programs written by the highest-rank students. The implementation of these strategies needs careful plan to not introduce new problems. For example, the equal-rank designation strategy may aggravate student polarization while the stepwise designation strategy requires a complex adaptive designation algorithm.

3. Adoption of anti-plagiarism mechanism. Successful anti-plagiarism techniques (Sun, Che, Wang, & Su, 2011) can be used in future system to effectively fight plagiarism and improve the assessment quality. It is also interesting to develop new incentive mechanism in student learning. To increase student learning motivations and achieve better learning outcomes, activities such as sample exhibition and code review competition may be used to offer students incentives for academic performance. The sample exhibition refers to displaying the best written program of each assignment (needing possible perfection.
by the teacher) on the website as a sample therefore student can learn from it and make reference to it. Since all students work on the same assignment and their memory is fresh, prompt sample exhibition will make positive influence on their competence in learning programing languages. The necessity of this activity has been indicated in a number of technical student interviews. Additionally, the activity of review competition can be held occasionally to trigger students’ interest in the PCR process and attract students to participate more actively in a programming course.

Acknowledgments

This research is supported by China Postdoctoral Science Foundation (gs1) (Grant No. 20100471042) and “985 innovation team” project of Harbin Institute of Technology (gs2).

Additionally, Dr. Michael Collins, an excellent Irish lecturer working with School of Computing, Dublin Institute of Technology (DIT), made careful proofread and proposed some valuable comments for this paper. We say thanks to him for his efforts.

References


