Quality Assurance of Peer Code Review Process: A Computer Science Based Strategy *

WANG Yan-qing[1], *YANG Fa*[2], *LIU Pei-jie*[3], *COLLINS Michael*[4]
(1. School of Software, Harbin Institute of Technology, Harbin 150001, China;
2. Shenzhen Beauphone Electronics Co., Shenzhen 518033, China;
3. Printing House, Harbin Institute of Technology, Harbin 150001, China;
4. School of Computing Dublin Institute of Technology, Dublin 8, Ireland)

Abstract: The software industry is asking universities and colleges to cultivate more software engineers who can write quality programs. A peer code review process is an ideal approach to maximize the learning outcome of students in programming. In this paper, the process in our previous publication was improved. The found problems were analyzed which will take as the basis of the future research on quality assurance. Finally, a set of solutions based on computer science were proposed to further improve the whole review process.
Key words: peer code review, quality assurance, management information system (MIS), software engineering education
CLC number: TP31 **Document code**: A **Article ID**: 0529-6579 (2007) S2-0116-05

1 Introduction

Since inspection was introduced as a means of software quality assurance, many software engineering educators have become more interested in introducing code review into their courses[1-3]. So far, the research on code review covers ① specific code review process or approaches[1-6], ②performance improving or quality assurance[7], and ③ supporting tools to enhance the learning outcome of code review[8]. Generally, the application of code review is divided into three categories[4-5]: self review[6], peer review and tutor review, among which peer code review (PCR) is recognized as the most practical one[5]. PCR is a technique which is generally considered to be effective on promoting students' higher cognitive skills[4], since students use their own knowledge and skill to interpret, analyze and evaluate others' work to clarify and correct it.

A PCR process presented in [5] was conducted on Year 2 students in our School of Software in Harbin Institute of Technology. Using this approach, students improved themselves when helping other students. Even though the students are the biggest beneficiaries, the burden on instructors was also lightened. According to the analysis of the extracted data, it was found that this PCR process has a high operability and thus a better learning outcome was achieved.

Nevertheless, after implementing this process during two academic years, i.e. 2005-2007 in an introductory programming class, some problems were found when checking the documents submitted by the students and also interviewing some students. Except for the problems associated with qualification issues such as some students lack the programming ability to write quality programs or to be qualified reviewers, the major problems come from the lacking support of management information system (MIS). The problems associated with student's qualification can be addressed by specific training on programming ability and reviewing technique. This paper thus focuses on quality assurance by process improvement and scientific management.

2 Peer code review process

So as to facilitate the communication with other researchers in future work, the peer code review process in [5] was refined in the following three aspects.

2.1 Redefinition of roles

Based on the popular naming convention in publication circle like journals or conferences, the roles in this PCR process are redefined as following. It is easy to find that, the first three roles are taken by students, while the last role is played by an instructor or a teaching assistant.

(1) Author is a student who accepts the code review activity by someone else;

收稿日期: 2007-10-28
作者简介: 王延青 (1968年生), 男, 讲师; E-mail: yanqing@hit.edu.cn

(2) Reviewer is a performer who does the code review activity;

(3) Reviser is the author himself/herself. When an author has accepted the form containing the comments and begins his/her revision work, he/she now change role and becomes a reviser;

(4) Instructor may be the lecturer or qualified teaching assistant[5] who is responsible for the programming lab class. They check the written code, review results, deliver grades to students, and tutor several students who are slowing in pace.

2.2 Redefinition of documents

Similarly, documents in[5] are improved. The detailed results are as follows:

(1) Manuscript code: source code the author has just completed. It is encouraged that the author reviews their own code before submitting, thereby making the following steps more efficient and productive.

(2) Comments form: the comments that the reviewer makes when they review the manuscript code written by the authors. As described in [5], the comments form checks for design defects, coding defects, code not complying with correct coding standards, and possible improvements, etc.

(3) Revision code: the final revised program after making changes to the manuscript code based on the received comments form. The quality of the revision code depends on the quality of the manuscript code, the reviewer's thoroughness, and the care of the reviser.

(4) Reference solution: the answer to the assignment and is held by the instructor. This should be the complete program. It will be used to assess the learning outcome of the author, reviewer, and even the reviser when required.

2.3 Refinement of PCR process

Besides the redefinition of roles and documents mentioned above, the whole process will be refined. Some phases are redefined more precisely and some are adjusted in time sequencing (Figure 1). In Figure 1, the two activities beginning with letter *S* belong to the submit phase and all the activities starting with letter *Q* fall into the quality assurance phase.

The refined PCR process consists of 6 phases which are made clearer than the definitions made in [5]. Each phase is described as follows:

Phase 1 3/ Write. A student (author) will complete their own assignment program (manuscript code) that may pass the compilation and running test (letter *W* in Figure 1).

Phase 2 3/ Submit. The author will email the manuscript code to the instructor as soon as possible

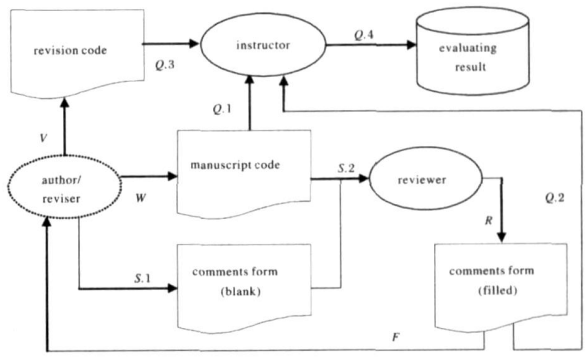

Fig. 1　PCR flow chart

(letter *Q*. 1 in Figure 1) to enable the instructor for to perform the quality management. Following this, the comments form with basic information such as author's name and reviewer's name (letter *S*. 1 in Figure 1) is sent to the specific reviewer (letter *S*. 2 in Figure 1) together with the associated manuscript code.

Phase 3 3/ Review. When the reviewer receives the manuscript code, they will perform the review as soon as possible. If any defects or possible improvements are found, suggestions or meaningful comments are entered carefully into the comments form (letter *R* in Figure 1) sent by author.

Phase 4 3/ Feedback. The reviewer sends the completed comments form to the author (letter *F* in Figure 1), and copies (cc) this to the instructor for quality management (letter *Q*. 2 in Figure 1).

Phase 5 3/ Revise. After receiving the comments form, the author makes revisions to their manuscript code (letter V in Figure 1) referencing the reviewer's comments and suggestions. The completed revised code will be emailed to the instructor for quality management (letter *Q*. 3 in Figure 1).

Phase 6 3/ Quality assurance. Having the manuscript code in *Q*. 1, comments form in *Q*. 2, and revision code in *Q*. 3, given sufficient time, it is not difficult for instructors or teaching assistants to check that the author and reviewer perform their work responsibly. Through instructor's checks (letter *Q*. 4 in Figure 1), it is easy for students' marks to be saved in a predefined database.

3 Problem finding

In the above PCR process, it is assumed that all the participants fulfill every phase carefully and responsibly. However, it was not very satisfactory. Through the Object Oriented Programming laboratory class of the nine students in the HIT-DIT joint program in spring semester 2007, it was found that the design of this PCR process was not perfect even though most of the

students received satisfactory marks in this course. This may have been a result of careless authors, irresponsible reviewers and busy instructors in the review process, which might discount the learning outcome of the whole PCR process.

The problems were discovered by observing the students' performance and interviewing with the students when they finished their academic year. These problems are summarized as follows:

(1) The process flow is difficult to control. Although students are asked to submit their work complying with the sequence described in Figure 1, a few of them did not submit on time. For example, students are required to email their manuscript code to the instructor for the quality control before submitting it to their reviewers. However, some of them submitted their manuscript code and revision code together at the last minute making it very difficult for the instructor to check if the manuscript code has not been changed when it is being reviewed. Another example is when all the reviewers are requested to send a copy of the comments form in time to the instructor when they give their feedback to their authors. However, a minority group of students often forget to do so. These problems make it impossible for the instructor to control the quality.

(2) Conspiracy exists. The fixed grouping strategy provides the opportunity for authors and reviewers, having a good relationship, to invent a method to blunder through. Case 1: After an author who may have a poor mark finishes his/her manuscript code, he/she sends the manuscript code to the reviewer in a non-procedural way. The reviewer performs the review and returns the comments to the author without recording it. The author might perform some update to the manuscript code before formally submitting it. Although some of the learning objectives can be met, the quality of the whole PCR can not be guaranteed. Case 2: The possibility exists where some reviewers finish the revision work instead of the revisers (authors). Case 3: Some authors write low quality programs and after negotiation with the reviewer, will agree to do a "shallow" review. Case 4: The reviewer completes the comments form very quickly after simply running the author's program and not reading the code carefully.

(3) Falsification of data. Falsification can be viewed as a specific case of conspiracy and is listed separately because of its high frequency. Sometimes when one of students in the same review group finishes the assignment program, their group mates might copy the written program and make some changes, such as adding some program comments or even make some negligible defects. After the falsification is done, the author's manuscript code enters the whole PCR process. This behavior causes a severe negative influence to emerge on the process assessment and its quality assurance.

Therefore, it is necessary to build up an appropriate control mechanism, such as constructing some software systems, to maximize learning outcome of all the game players.

4 Quality Assurance Strategy

The improvement and utilization of technology can often change the result of management, and the PCR process is not an exception to this. Some MIS based on computer science and technology can not only decrease the working intensity of participants, but also make some improbable management patterns practical. With the support of the MIS, the PCR process in this paper will be more successful.

4.1 Web-based PCR management information system

It was found that time sequence problems exist in the current PCR process. The timing problems are often the key factors affecting the quality assurance of PCR. For example, before the well developed information system commences, the common way for instructors and students to transfer files is via email. As is known, the delay or error of email sometimes becomes some students' excuse for not submitting their assignment work on time. In fact, this problem does harm the students' time management responsibilities as well as impede the development of the instructor's quality management.

As well as this, a web based PCR management information system is under construction. In this system, a blind review mechanism will be carried out by a computer and strict submission, reviewing, and revision management can solve the quality assurance problems caused by time sequence issues. For example, when an author submits the manuscript code, a copy will be automatically sent to the instructor by the system. When a reviewer gives the comments form back to the author, a copy of comments form will be sent to the instructor. These procedures can prevent situations such as delayed submission, repeated submission and missed submission. Most importantly, the MIS can simplify the reviewing process. For example, the blank comments form in Figure 1 can be generated automatically by the computer, which can reduce the author's workload and the difficulty of implementation. When the MIS is completed, the process in Figure 1 could be further simplified and consists of four main phases: write/submit,

review/feedback, revise/finalize and quality assurance (see Figure 2). The web based PCR process is more effective and convenient for student learning, which will elevate the probability of successful implementation of PCR.

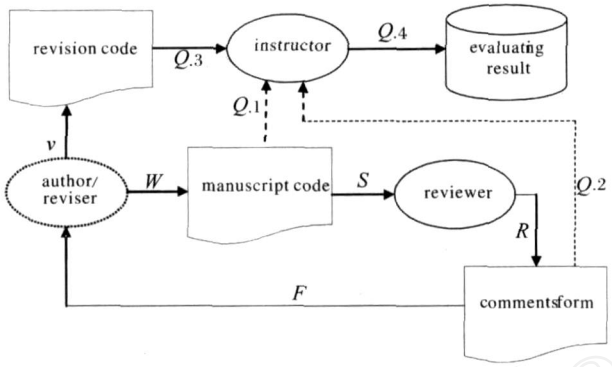

Fig. 2　PCR process supported by MIS

4.2　Evaluating system for coding standards

After the recent publication[8], lots of research and development work about coding standard was undertaken and an online evaluating system was developed. With the open web based MIS, the instructor can set a threshold value and make sure that only the source code which can pass the evaluation can enter the review process. This approach can prevent the unqualified or careless student from submitting poorly written code.

In fact, with the help of coding standard evaluation, this approach takes complying coding standard as the prior condition of entering the formal code review procedure. The advantage of this strategy is that it makes the reviewers concentrate their mind on the problems beyond coding standards so that the learning outcome can greatly increase. However, the student must be well qualified to find high level and complicated errors rather than coding standard issues. So this approach can be applied in high level students, otherwise the threshold must be adjusted lower for students learning introductory programming language.

4.3　Code review tool

Early in 1996, Belli and Crisan presented the automation of code review[9]. Later on Jun-Suk Oh and Ho-Jin Choi demonstrated the automated code review prior to manual code review[10]. Also, a static analysis tool targeting lightweight program verification and finding coding defects in Java was developed[11].

On the basis of the above references, the authors of this paper believe that some obvious defects can be removed before manual review by some tools based on computer science. For example, simulated compiler techniques can help find simple syntax errors, while a logic error checker can help filter specific and obvious logic errors. The suggested strategy is that all the programs which can not pass these tools are not allowed to enter the PCR procedure.

4.4　Revision management

The difference between revision code and manuscript code is the key to the quality assurance of PCR, the key to evaluating whether an author deals with their study carefully, and the key to assess the outcome of the whole learning process. An author must finish the revision code based on the manuscript code and there should be a difference between them. The difference depends on the comments form and the reviser's behavior. In fact, if the source code difference analysis tools like in[12-13] are developed, then the computer can do the comparison of two versions instead of the instructor. What the instructor needs to do is just check whether the difference is in-line with the comments form. If this approach is successful, then the efficiency of the whole PCR process will greatly enhance.

With the support of MIS mentioned above, the PCR process will become more practical and efficient. While the objective to enhance the whole learning outcome is achieved, the quality of reviewing gets adequate assurance and instructors can value their time for an effective implementation and scientific management of the PCR process.

5　Conclusion and Future Work

When the software industry pays more attention to quality management and begins to research how to control the high cost of testing and maintenance, the code review at source code level attracts more and more emphasis. Even though code reviews are time consuming, they are much more efficient than testing. A typical engineer, for example, will find approximately 2 to 4 defects in an hour of unit testing but will find 6 to 10 defects in each hour of review code[6]: 160.

However, the process in our previous publication[5] is not perfect. Some problems were found after the implementation in two academic years, so a series of further research was undertaken. They are: ① many improvements were done from the view of role, documentation and procedures to make the PCR process clearer and more convenient for future research; ② on the basis of laboratory observation and interviewing with students, it was found that there are obvious process management issues such as time sequencing which can determine the fate of this PCR process; ③ the constructive solutions based on computer science to implement the PCR process successfully were put forward.

At the same time, there is lots of future work.

First, the PCR process is recognized as a game theory issue, so it is necessary to build up a game theory model to control the behavior of all participants. Second, the management information system mentioned above will be a main part of our future work, especially the information system based on web services. Last, if a source code library could be built, in which every sample has been checked and analyzed quantitatively and objectively, the ability of students to do code reviews can be measured more precisely. By this meaning, some born testers could be discovered[7] who are presently lacking in most countries.

References:

[1] GEHRINGER E F, CHIN D D, PEREZ-QUINONES M A, et al. Panel: Using Peer Review in Teaching Computing [C]. Proceedings of the 36th SigCSE Technical Symposium on Computer Science Education, 2005, 37: 1.

[2] SITTHIWORACHART J, JOY M. On-line Feedback and Assessment: Effective Peer Assement for Learning Computer Programming [C]. Proceedings of the 9th Annual SigCSE Conference on Innovation and Technology in Computer Science Education, June 2004.

[3] TRYTTEN D. A Design for Team Peer Code Review [C]. Proceedings of the 36th SigCSE Technical Symposium on Computer Science Education, 2005, 37: 1.

[4] FALLOWS S, CHANDRAMOHAN B. Multiple Approaches to Assessment: Reflections on use of tutor, peer and self assessment [J]. Teaching in Higher Education, 2001, 6(2), 229 - 246; Association for Learning Technology Journal (ALT-J), 2001, 9(1): 26 - 37.

[5] WANG Yan-qing, SU Xing, HU Yan, et al. How to Evaluate Students' Learning Outcome: A Peer Code Review Model in Undergraduate Computer Programming Class [C]. Proceedings of the 2nd International Conference on Computer Science and Education. Wuhan, China. 2007: 1292 - 1295.

[6] HUMPHREY W S. Introduction to the Personal Software Process [C]. Addison-Wesley, Pearson Education, Inc. 2002: 159 - 163.

[7] CUNHA Alessandra Devito Da, GREATHEAD David. Does Personality Matter? An Analysis of Code-review Ability [J]. Communications of the ACM, 2007, 50(5): 109 - 112.

[8] WANG Yan-qing, WANG Jian-zheng, SUI Xiang-nan, et al. Quantitative Research on How Much Students Comply with Coding Standard in Their Programming Practices [C]. Proceedings of the 3rd China Europe International Symposium on Software Industry Oriented Education, Dublin, Ireland. 2007: 116 - 119.

[9] BELLI, F, CRISAN R. Towards automation of checklist-based code-reviews [C]. Proceedings of Seventh International Symposium on Software Reliability Engineering. 30 Oct. -2 Nov. 1996: 24 - 33.

[10] OH Jun-Suk, CHOI Ho-Jin. A Reflective Practice of Automated and Manual Code Reviews for a Studio Project [C]. Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS '05). 2005: 37 - 42.

[11] GEAY Emmanuel, YAHAV Eran, FINK Stephen. Continuous Code-Quality Assurance with SAFE [C]. PEPM '06 January 9 3/ 10, Charleston, South Carolina, USA. 2006: 145 - 149.

[12] MALETIC J I, COLLARD M L. Supporting source code difference analysis [C]. Proceedings of 20th IEEE International Conference on Software Maintenance, 11 - 14 Sept. 2004: 210 - 219.

[13] COLLARD M L, KAGDI H, MALETIC J I. Factoring Differences for Iterative Change Management [C]. Sixth IEEE International Workshop on Source Code Analysis and Manipulation (SCAM'06). 2006: 217 - 226.