Preprint. Use following information for referencing: Hämäläinen Harri, Tarkkonen Jukka, Heikkinen Kari, Ikonen Jouni and Porras Jari: Use of Peer-Review System for Enhancing Learning of Programming. The 9th IEEE International Conference on Advanced Learning Technologies, Riga, Latvia, July 14-18, 2009. pp.658-660, ISBN 978-0-7695-3711-5

Use of Peer-Review System for Enhancing Learning of Programming

Harri Hämäläinen, Jukka Tarkkonen, Kari Heikkinen, Jouni Ikonen and Jari Porras
Lappeenranta University of Technology
P.O. Box 20, FIN-53850 Lappeenranta, Finland
Firstname.Lastname@lut.fi

Abstract

In this paper we present a study where a peer reviewing system targeted for conference paper evaluation has been used for enhancing the learning of programming though peer review. For peer reviewing purposes two open source systems were evaluated and MyReview was selected for evaluation. The peer review system was used in one programming focused course and the benefits of peer reviewing were analyzed from the reviewer and reviewee point of view. The results show that a peer review system targeted for conference paper evaluations is applicable for reviewing programs although the system should be much simpler. Students appreciated the documented comments from other students but were more interested in lecturer or other expert opinion.

1. Introduction

Programming is one of the key skills that a computer science student must learn. Several different learning approaches have been used for programming. In general, the learning process of programming does not differ from basic learning principles. Feedback is important to make the students aware of how they are doing. Understanding basic ideas of programming and given task is important, so the students understand what to do and to know how to find more information.

Feedback can be provided in various ways; a) teachers can assess the student's work, b) programs can be automatically evaluated e.g. by an online service, or c) assessment can be executed by other students. In programming tasks this is not usually as simple as there might not be just one correct answer and the programming styles may vary. However, co-student may give a new idea how to solve the problem. This way both learn from each others' solutions. If this kind of a practice becomes common, students might also consider how readable they make of their programs, as they know that other people will be reviewing it. In this paper we consider peer review process and the experiences of a web-based MyReview application.

The paper is organized as follows. After the introduction some peer review approaches and practices from literature are presented. Chapter 2 focuses on the peer reviewing process used on our experiment. Chapter 3 presents the results of the peer reviewing and in Chapter 4 conclusions are presented.

1.1. Related work

Peer review is widely studied in different contexts. However, for the purpose of this paper we took a look at the papers that focused on programming and web-based peer reviews.

Wang et al. [1] utilizes Peer Code Review (PCR) process in maximizing learning and at the same time providing quality code. They introduce a PCR process through which students gain an enhanced capability to provide peer assessments to others and conducting quality coding. The PCR process consists of six phases; firstly the student (author) completes his/her programming task, secondly he/she submits the code to instructor and to reviewer (another student), thirdly the reviewer inserts his/her comments, fourthly the reviewer sends these comments both to the author and instructor, fifthly the student may revise his/her code based on the comments of the reviewer and sends this to the instructor and sixthly the instructor evaluates the code.

Denning et al. [2] introduce in-class lightweight preliminary peer-review (LPPR). In their approach, peer review is already started in the classroom so that the

Preprint. Use following information for referencing: Hämäläinen Harri, Tarkkonen Jukka, Heikkinen Kari, Ikonen Jouni and Porras Jari: Use of Peer-Review System for Enhancing Learning of Programming. The 9th IEEE International Conference on Advanced Learning Technologies, Riga, Latvia, July 14-18, 2009. pp.658-660, ISBN 978-0-7695-3711-5

students classify the solutions into four dimensions, i.e. correctness, comprehension, worthiness for discussion and similarity to evaluator's own solution. Their approach consists of a process and a supportive tool. Their LPPR has five key components, a) quick sharing on in-class student generated code, b) lightweight selection review criteria, c) prompt self-reflection, d) use of peer reviews, and e) instructor overview of responses.

Liu et al [3] introduce a web-based peer review as a learning strategy. The strategy was built to support student's learning so that the accompanying study could provide answers to questions. The results indicated that students performed better while using peer review, clear majority (67%) preferred the peer review, metacognition levels were high (77%) and correlations to learning were significant. However, these results were experimental as no control group was used. Another web-based peer-review and grading system is introduced by Gehringer [4] who also points out that the students have benefited of using peer review. Carlson et al. [5] also utilize computer-mediated peer review. They utilized Calibrated Peer Review for trying to improve the quality of written reviews. Coit and Stowe [6] present web-based peer review tool that is built to be learner-centric. It is especially suitable for community-oriented activities.

Crespo et al. [7] introduce an adaptive strategy for peer review. In their strategy the peers, who evaluate, are matched against each others profile. The purpose is that by matching the quality of peer review can be enhanced. The matching algorithm includes correctness of pairs and optimization goals (e.g. reliability, pedagogical understanding).

2. Evaluation of code peer review

As peer reviewing has been seen in the literature a beneficial act for the students we wanted to experiment the peer reviewing with our students. The peer review process used for this study consisted of the following steps:

1. Discussion with the lecturers to find out the requirements for the peer reviewing system
2. Selection of the peer review system based on the requirements and setting up the system
3. Usage of the Peer review system
   a. Submitting finished/ready program codes into the system
   b. Reviewing other group's program
   c. Getting feedback of own code
4. Evaluation of the benefits of peer reviewing

2.1. Course details and requirements

The peer review system was tested in *Concurrent and parallel computing* course, a Master's level course focusing on the understanding of concurrency and parallel computing principles. All the 21 participating students had at least moderate background on programming (by having 3-5 programming courses). The course has two separate practical programming tasks of which only one task was peer reviewed while the other task was used as a reference. The challenge in this course is not so much in programming technique but in the understanding of the parallel computing principles. Practical programming tasks emphasize this understanding and the programming tasks are selected in such a way that there exist several acceptable solutions for the problem, e.g. sorting problem.

2.2. Peer review process in the selected course

Before the peer review was performed, a suitable peer reviewing system needed to be selected and methods for utilizing it to be installed. In the first phase some 20 systems containing peer review features were evaluated against the preset requirements. Most of the peer review systems available fulfilled the requirements and therefore more weight was given for criteria like open-source and freeware. Finally two systems, OpenConf and MyReview, were selected for the final comparison. As the laboratory had some previous experience with MyReview and there were only minor differences between the systems, MyReview was selected.

The usage of the peer review system was divided into three parts: Submission, peer review and feedback evaluation. The peers for the submitted programs were selected by the lecturers based on the solutions and social distance of the students.

Each group was given a deadline to upload their programs that were then set up to the review system. Peer review phase required students to download the program package, to setup the program into computing environment and to run the program. After getting everything to work they were supposed to look at the actual code and evaluate the following aspects of the program: a) use of dynamic processes, b) communications, c) parallelization aspects, d) memory

Preprint. Use following information for referencing: Hämäläinen Harri, Tarkkonen Jukka, Heikkinen Kari, Ikonen Jouni and Porras Jari: Use of Peer-Review System for Enhancing Learning of Programming. The 9th IEEE International Conference on Advanced Learning Technologies, Riga, Latvia, July 14-18, 2009. pp.658-660, ISBN 978-0-7695-3711-5

handling, e) clarity of code and f) execution time (compared to their own code). In addition to the numerical values the students were asked to give free comments (clarification for their grading) to each of these aspects. These comments were later used for evaluating the quality of the work the students did.

After the peer review phase each group received the comments to their own task. Students were then asked to consider the given proposals as proposed in PCR process but were not forced to do any changes. Since the workload of programming tasks was moderate, all the components of LPPR were not considered feasible in this case.

3. Results of the evaluation

The results of the peer review experiment can be evaluated from two perspectives: a) suitability of the selected system and b) quality of the peer review. None of the students had completed this kind of a peer review before.

Even if the system was not modified at all for code reviewing purposes, half of the respondents felt that the peer review process was easy enough. However, grouping students and user account management as well as organizing information were not efficient for code reviewing. As MyReview is distributed under the GNU General Public License [8], the source code can be modified. This will give a chance to make the system more suitable for peer reviewing of source code.

Table 1 presents a comparison of the lecturer and the student based reviews. As the students used different scale for grading the given numerical values were normalized (LWeight and PWeight) before comparison. The *difference* column shows the difference between evaluations. In order to evaluate the quality of the evaluation the free text *comments* were analyzed and graded into levels A to C based on the comment quality and quantity (A meaning excellent and C for poor comments). Shading shows the best works and smallest differences.

Table 1. Comparison between lecturer grading and student review (LPoints = Lecturer grading, PPoints = Peer grading, (L/P)Weight normalized grading, Difference = PWeight-LWeight, Peer = group that performed review, Comments = quality of the comments)

Group	LPoints	LWeight	PPoints	PWeight	Difference	Peer	Comments
1	9,50	79,17	-	-	-	-	-
2	3,50	29,17	2,58	36,90	7,7	1	A
3	11,50	95,83	5,42	77,38	-18,5	4	A
4	7,50	62,50	5,17	73,81	11,3	6	A
5	4,50	37,50	5,00	71,43	33,9	3	C
6	11,00	91,67	4,83	69,05	-22,6	8	B
7	4,00	33,33	1,00	14,29	-19,0	11	C
8	8,00	66,67	4,42	63,10	-3,6	7	A
9	10,50	87,50	6,50	92,86	5,4	5	A
10	5,50	45,83	2,67	38,10	-7,7	2	B
11	10,00	83,33	6,42	91,67	8,3	9	A
Average		64,77		62,86	-0,48		

It seems that the average weight of the evaluations is almost equal (64,77 and 62,86). However, only half of the evaluations match nicely (difference <10) while in the rest there exists huge differences. All nicely matching evaluations were done with proper quality (based on the quality of comments). It is notable that the two best solutions (by the lecturer) did not get high peer review grading.

The direct value of the peer review for the students can be divided in two parts. Firstly, the feedback and ideas for improvements that are got from the other groups and secondly the benefit of seeing other types of solutions while reviewing other groups' codes.

4. Conclusion and future work

In this paper we have presented a study of peer reviewing for the enhancement of programming learning. The attitude of the students to use the peer review was fairly positive. We believe that the more advanced students are able to appreciate the positive aspects for their personal learning and knowhow that giving and receiving peer reviews can create. The correlation between students' and teacher's votes were weak and this requires extra attention in the future.

To make the peer review comfortable, the software used the for peer review process should be formed to more student friendly direction. Now the software is purely meant for a conference paper review and it is not flexible enough for different programming assignment evaluation purposes. We believe that identifying the most distracting problems and by correcting these, usability of this application for peer reviewing increases considerably.

5. References

[1] Wang, Y., Li Y., Collins, M. and Liu, P., "Process improvement of peer code review and behavior analysis of its

Preprint. Use following information for referencing: Hämäläinen Harri, Tarkkonen Jukka, Heikkinen Kari, Ikonen Jouni and Porras Jari: Use of Peer-Review System for Enhancing Learning of Programming. The 9th IEEE International Conference on Advanced Learning Technologies, Riga, Latvia, July 14-18, 2009. pp.658-660, ISBN 978-0-7695-3711-5

participants", *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, Portland, OR, USA, 12-15th March 2008.

[2] Denning, T. et al., "Lightweight preliminary peer review: does in-class peer review make sense?", *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, Covington, Kentucky, USA, 7-10th March 2007.

[3] Liu E., Chiu, C. and Yuan,S., "Web-based peer review: the learner as both adapter and reviewer", *IEEE Transactions on Education*, Volume 44, Issue 3, Aug. 2001, pp. 246-251.

[4] Gehringer, E.F., "Strategies and mechanisms for electronic peer review", *30th Annual Frontiers in Education*, Volume 1, Kansas City, Missouri, USA. 18-21st Oct. 2000, pp. F1B/2 - F1B/7.

[5] Carlson, P.A.; Berry, F.C.; Voltmer, D., "Incorporating student peer-review into an introduction to engineering design course", *35th Annual Frontiers in Education*, Indianapolis, IN, USA, 19-22th Oct. 2005, pp. F2C-20-5.

[6] Coit, C.; Stowe, K., "Peer Review for Life", *Sixth International Conference on Advanced Learning Technologies*, Amsterdam, The Netherlands, 5-7th July 2006.

[7] Crespo, R.M.; Pardo, A.; Kloos, C.D., "An adaptive strategy for peer review", *34th Annual Frontiers in Education*, Savannah, GA, USA, Volume 2, 20-23rd Oct. 2004, pp. F3F-7-13.

[8] The MyReview System, Website. Available at: http://myreview.lri.fr/. Accessed: January 14th 2009.