

COMMUNITY DIRECTIONS

Transition from Passive Learner to Critical Evaluator through Peer-Testing of Programming Artefacts

Gudmund Grov, Mohammad Hamdan, Smitha Kumar, Manuel Maarek*, Léon McGregor, Talal Shaikh, J.B. Wells, Hind Zantout

Edinburgh (UK) and Dubai (UAE) campuses, Heriot-Watt University

*Corresponding author: M.Maarek@hw.ac.uk

Keywords: *Computer science education, peer testing, peer feedback, software testing, student transitions*

Abstract

Offering timely feedback on programming while encouraging learners to engage in critical evaluation of programs are the objectives of peer-testing. We report on a peer-testing experiment with students on distant campuses using a Web platform. The experiment shows the potential that peer-testing has to help students transition from passive learners to critical evaluators.

Introduction

The teaching of programming in Computer Science courses has an important practical element in which learners are usually expected to implement a piece of software matching a given specification (project-based learning). Teachers give feedback on and evaluate the quality of the program code and the compliance of its behaviour to the given specification. A number of issues arise from assessing programming and providing feedback on such coursework.

- The focus of most students is often on the development side of the coursework, and less on the tasks intended to train the students' evaluation skills.
- Producing quality feedback and evaluating students' programs is a time consuming activity; as a result feedback on programming coursework often comes too late for the students to revisit the code they wrote.

- Students value peer-interaction and group work but there is a need for individual programming coursework in order to assess individual programming skills Properly.

Peer-assessment is known to improve assessment and to positively impact students' learning (Falchikov 2013; Mulder, Pearce & Baik 2014; Nicol, Thomson & Breslin 2014). An experiment in peer-review to teach software testing (Smith et al. 2012) showed the positive impact in raising students' perceptions of testing. We propose Peer-Testing: peer-feedback via testing of programming artefacts. Peer-testing involves learners, allows for more timely feedback and enhances the students' critical analyses of programs. One aim of peer-testing is to help consolidate the programming skills of students by making them engage in critical analysis of programming. Through peer-testing students need to actively test their peers' programs and identify the cause of a failed test to give informative feedback. In doing so they should engage differently with the specification than they first did when developing their own programs. As a peer activity, the goal of peer-testing is to make the learners critically analyse their own practice by assessing their peers. Peer-testing is therefore a tool for helping the student transition from programming as a university-based individual activity to a professional practice where

collaboration and peer-review are key for employability.

We have developed a Web platform for peer-testing (Maarek & McGregor 2017), the design of which was informed by interviews of programming teachers in Higher Education. We evaluated the platform and the peer-testing approach with an experiment with learners from the two Heriot-Watt University campuses where our computer science programmes are taught. The evaluation was through a questionnaire and a focus group discussion with learners.

We report here on this experiment giving insights as to how peer-testing and our Web platform can impact the teaching of programming. The main outcomes are: (1) learners practice and build testing habits with peer-testing; (2) learners value the benefits of exchanging feedback on programming, although they agree that it should not replace teachers' feedback; (3) learners value peer-testing as a cross-campus learning activity and suggest it enables indirect learning from teachers of the other campus. Peer-testing could therefore participate in preparing the students for a workplace where programming is more collaborative (points 1 and 2) and where teams may be scattered around the globe (point 3), a scenario that is common nowadays.

The work presented here took place at Heriot-Watt University during Léon McGregor's BSc Honours project (McGregor 2017) and as part of the University's Learning and Teaching Enhancement project lead by Manuel Maarek. It was presented at the Horizons in STEM Higher Education Conference 2017. Aspects of the project, in particular the development process of the Peer-Testing Web Platform will be presented at the PLATEAU 2017 Workshop (Maarek & McGregor 2017).

Peer-Testing using the Web Platform

The experiment we conducted was to evaluate the peer-testing approach and the

Peer-Testing Web Platform we have developed. This web platform provides an online facility to organise and run peer-testing sessions with students, where students will prepare a solution (program answering the problem set by the coursework) and test cases (tests which verify that a program behaves as intended by the specification). A peer-testing session is analysed into the following consecutive stages. The teacher sets up individual coursework tasks and could provide sample black-box solutions and sample tests (Stage 0, set-up). The students can upload their solutions and their test cases. They can run theirs or the teachers' solutions against their own tests or the tests provided by the teacher (Stage 1, self-testing). The teacher can then move the system to peer-testing mode where students (in small groups) could access and run the solutions and tests of their peers; they are also invited to discuss the running of each test case directly through the discussion interface of the platform (Stage 2, peer-testing). Finally, the students are invited to submit to the teacher a reflective report of their involvement in receiving and giving feedback through tests (Stage 3, reflection). Figure 1 shows the page that outlines to the students the different artefacts for a given coursework, while Figure 2 displays the discussion occurring around a solution and a test result. For more details about the features of the Peer-Testing Web Platform, see (Maarek & McGregor 2017). The source code of the Web platform is available on GitHub (<https://github.com/peergramming/peer-testing>).

Peer-Testing Experiment

In order to evaluate peer-testing and our Web platform, we invited participants to work on a small programming coursework. Upon completing it, they were given a questionnaire to fill out and were invited to a focus group discussion session via video-conference, see Figure 3.

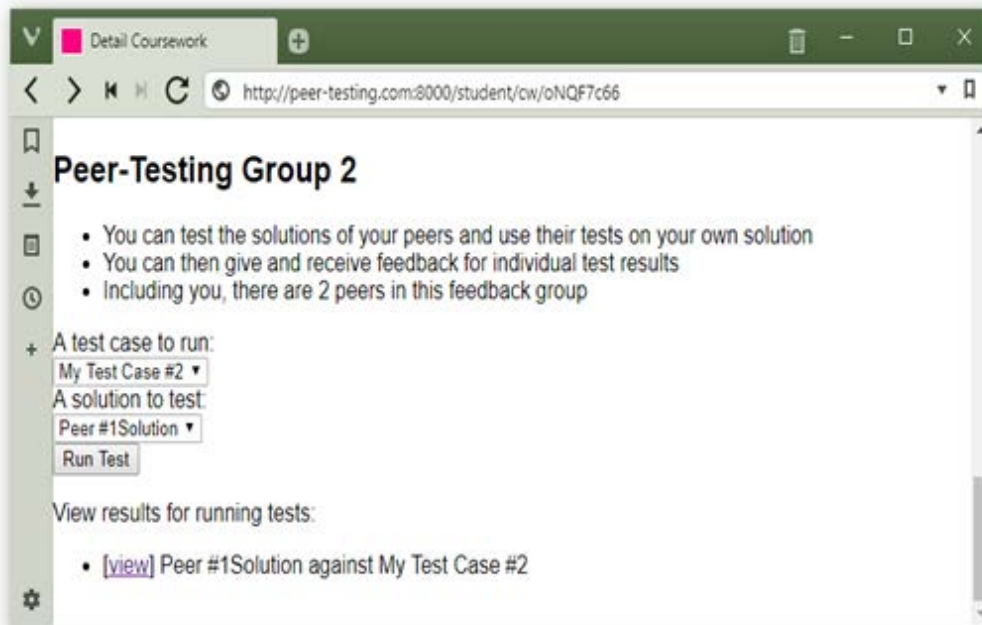


Figure 1 Screenshot of the Peer-Testing Web Platform's test running page.

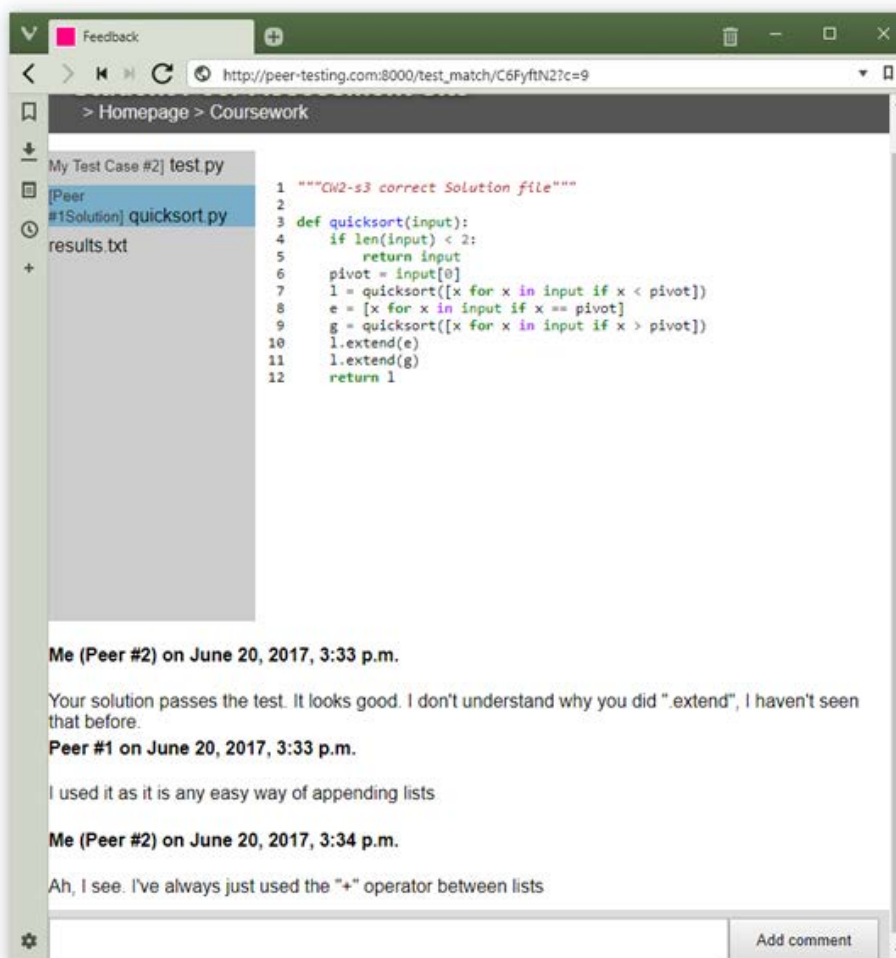


Figure 2 Screenshot of the Peer-Testing Web Platform's feedback page



Figure 3 Cross-campus focus group discussion via video-conference.

Participants to this research experiment were volunteers who were recruited from the 2nd and 3rd year cohorts of the Computer Science department of Heriot-Watt University in the Edinburgh and Dubai campuses. Overall, 12 participants used the Web platform and 11 filled out the final questionnaire. The demographics for this final questionnaire were as follows.

- Most of those who took part were Year 2 students, 7 in total, and 3 from Year 3 (with 1 participant not identifying the year of study in the form).
- There was a decent mix of students from both campuses, with 5 participants from the campus in Edinburgh and 6 in Dubai.
- There were more male participants, with 7 identifying themselves as male, and 4 as female. (Note that gender groups in the cohorts are less balanced than in our participants)

We believe this spread of students who took part provided a reasonably diverse range of participants.

Perceptions of Peer Activities

In order to find out students perceptions of peer activities, we asked the participants if they had already experienced peer-assessment or peer-feedback. Only 2 respondents to the survey had prior experience in peer activities. The peer

activities were “Essay peer-marking, giving feedback” and “Checking correctness and logic of code of other students in high school”. These responses would seem to suggest that peer activities are rarely utilised in the context of programming courses at University level.

The participants were asked to rate their opinions on whether or not they agreed with certain website and peer-testing related statements. The responses are given in Table 1.

It is reassuring to see that the results confirm the previous research into this area, suggesting that students believe they learn more by giving feedback as opposed to receiving it. There seems to be a somewhat mixed response to the idea that peer-testing can be better than a teacher's response, so an addition the platform may benefit would be from allowing teacher interaction as well as peers. For the most part, it seems that students respond positively that the peer-testing helps with completing the exercises, and that this should be encouraged in the university. Interestingly, the response to using such a peer-testing web platform more in university was less positive. This may be an indicator that the participants did not find the website all that useful (the web platform is in development; the version used for the experiment is a research prototype with a minimalist user interface).

Receiving feedback from peer-testing helps me learn	5	5	5	5	5	5	5	4	4	4	3	3
Giving feedback during peer-testing helps me learn	5	5	5	5	5	4	4	4	4	4	4	3
Peer-testing is better for learning than having a teacher give feedback on my work	5	4	4	4	3	3	3	3	3	3	2	2
Peer-testing helped me understand how code testing works	5	5	5	5	4	4	4	3	3	3	2	1
Peer-testing helped me understand how code review works	5	5	5	5	4	4	3	3	3	3	3	1
Peer-testing helped me understand the coursework exercises	5	5	5	4	4	3	3	3	3	3	3	1
Peer-testing helped me think differently about the coursework exercises	5	4	4	4	4	4	3	3	3	2	2	1
Peer-testing helped me think of new solutions to the coursework exercises	5	5	5	4	4	4	4	3	3	2	2	2
I think peer-testing exercises should be included as part of university courses	5	5	5	4	4	4	4	4	4	4	4	2
Peer-testing exercises using a website should be included as part of university course	5	5	4	4	4	4	3	3	3	3	2	2

Table 1 Responses for Likert questions "Do you agree..." with regards to peer-testing and the website (5 – Positive, 1 - Negative)

Peer-Testing within Courses

Having had some experience with the website, we asked the participants in the focus group discussion if they felt there were any courses or activities within courses which would be improved by using the peer-testing feedback offered by the website. They suggested that the website would be useful for:

- Larger pieces of coursework, which are likely to have very different solutions thus offering more places for discussion;
- The course on data structures and algorithms, where there are specific assignments that fit well with the website;
- Any programming exercise where the interface and specification is clearly set out.

One concern was raised however in that offering such peer-testing might be detrimental to completing some exercises as it opens up a surface for plagiarism. (This was a concern with the prototype used during the experiment, the Web platform now operates with two modes, requiring solutions to be uploaded before feedback is given.). But in contrast, the point was made that there is a big difference in allowing such feedback to be given before the final submission deadline, as it would allow students to improve their code based on the feedback given. A suggested middle-ground was that there be three deadlines for coursework – one for the submission of a solution, one for giving feedback, and one for suggesting how code would be fixed based on that feedback – where the overall grading for the coursework

took stock of each point of participation (this corresponds to the stages outlined earlier).

It was noted during the discussion that when a student is giving feedback to a peer, it is much more direct and that you can share “insider knowledge” of shortcuts in programming through discussion and feedback.

Unit Testing

During the discussion, a point was raised about how people normally do testing. Some said that they did not always do formal testing, but informally tested their code as it was developed. One participant said that by forcing themselves to write the unit tests this time, something they did not normally do, it revealed shortcomings in their own code that they had missed. This would suggest that the way the site forces the students to write code is beneficial to their self-testing and to their practice of software testing.

Cross-Campus Learning

One of the benefits that using a website would offer is that it could make it easier for peer-testing activities to take place across campuses. During the discussion session, when the participants were asked about their opinions on cross-campus learning such as the one of this experiment, they felt that it was much more “interesting”. They noted that within the feedback, there is greater scope for learning across campuses as you can give feedback based on different experiences you have had learning, and the different ways that lecturers will teach across campuses.

Conclusion and Future Works

The website did seem to make some steps towards providing better feedback and learning. The way the website operates allows for cross-campus activities and forces students to make use of unit-testing. Both of these features help enhance the learning experience and give opportunities for feedback. Students felt that peer activities would be welcome if they were used in university courses.

After this experiment, we are planning to use the Peer-Testing Web Platform in a programming course. We expect that experiencing the use of the peer-testing method will give us insight into how to further improve both the method and the web platform for effective learning.

Acknowledgements

This work was in part supported by a Heriot-Watt University and QAA joint funding.

References

Falchikov, N., (2005). *Improving Assessment Through Student Involvement: Practical Solutions for Aiding Learning in Higher and Further Education*. Routledge.

Maarek, M. & McGregor, L., (2017). *Development of a Web Platform for Code Peer-Testing*, in: The 8th Workshop on Evaluation and Usability of Programming Languages and Tools (PLATEAU) at SPLASH 2017. To appear online.

McGregor, L. (2017). *Web Platform for Code Peer-Testing (BSc Honours)*. Dissertation, Heriot-Watt University.

Mulder, R.A., Pearce, J.M. & Baik, C. (2014). *Peer review in higher education: Student perceptions before and after participation*. Active Learning in Higher Education. DOI: 10.1177/1469787414527391

Nicol, D., Thomson, A. & Breslin, C. (2014). *Rethinking feedback practices in higher education: a peer review perspective*. Assessment & Evaluation in Higher Education 39, 102–122. DOI:10.1080/02602938.2013.795518

Smith, J., Tessler, J., Kramer, E. & Lin, C. (2012). *Using Peer Review to Teach Software Testing*, in: Proceedings of the Ninth Annual International Conference on International Computing Education Research, ICER '12. ACM, New York, NY, USA, pp. 93–98. DOI: 10.1145/2361276.2361295