

# Method for Social Programming and Code Review

Michal TOMLEIN\*

*Slovak University of Technology in Bratislava  
Faculty of Informatics and Information Technologies  
Ilkovičova, 842 16 Bratislava, Slovakia  
michal@tomlein.org*

In order to create high quality, reliable and reusable code, incorporating code review into the development process is amongst the most effective of the available options. Due to its effectiveness in ensuring the quality of software, code review has seen wide acceptance in the industry.

However, in addition to being an important quality assurance method, it is also a powerful learning tool. We believe it can effectively be used to significantly improve the outcomes of the learning process and its overall quality. Furthermore, its adoption in the learning process in the form of peer reviews can serve to prepare students for code review in development practice, which is also highly desirable.

In recent years, social approaches to software development have changed the way we look at code sharing, collaboration and the development process. With the advent of social programming and code sharing services such as GitHub, there have been sweeping changes to the way we perceive and expect development of open source software components to work. Unfortunately, social programming has not made its way into programming courses to the same extent. In our work, we intend to combine the benefits of both code review and social programming, to improve the quality of the courses to provide additional development skills.

In many cases, peer reviews have been found to be adequate substitutes for pedagogical reviews. Peer reviews do, however, present a few problems in their realisation. Students' ability to review code and provide useful feedback varies to the degree that it becomes very important to select the most suitable reviewers for the individual problems the students have.

Reviewer selection is a non-trivial problem. The relevance and quality of the resulting review is highly dependent on reviewer selection, as different reviewers may have a different amount of experience with the particular problem they are assigned to review. Research has shown that reviewers with insufficient knowledge and experience in the problem area only contribute to user confusion and do not provide the necessary

---

\* Supervisor: Jozef Tvarožek, Institute of Informatics and Software Engineering

motivation [2]. Because of this, there is a need for a solution, which takes such reviewer attributes into consideration when selecting the most suitable reviewer.

Our approach is based on real-time code review using a shared live view of the code being reviewed, which the reviewer can see and comment on. This is in contrast with more traditional code reviews, which are asynchronous and therefore do not require the reviewer to be available at the same time as the author of the code.

Secondly, the method incorporates aspects of social programming to create a sense of community within the programming course. The intent is to raise the students' awareness of the whole group's progress and their place within the group using progress visualisation and to encourage them to participate in code review to help their fellow classmates complete all of the programming assignments.

We are currently evaluating the outcomes of an initial experimental study. We have collected data on 172 students in an introductory programming course at the faculty. The data contains observations of student work and describes the relationship between the personality traits and characteristics and the reviewing abilities. We designed the experimental study as an ordered sequence of 5 highly interdependent programming assignments on introductory cryptanalysis. The functionality necessary to support the live reviews and automatic reviewer assignment was built into the existing web-based learning platform deployed within the course, called Peoplia.

Using the data obtained from our experiments, we intend to determine and employ the correlation between personality traits and the abilities to deliver and receive help to be able to select a suitable reviewer with the highest probability of success in helping a particular user.

By exposing the students to other students' code through peer code review, we aim to inspire them to improve their own code, their ability to read and understand other code, to learn about different ways of looking at the same problem, and last but not least, to train them to be able to provide feedback, which is in itself an exercise of learning by teaching. Learning by teaching is a well-known practice effective especially in the long term, as it prepares students to learn new concepts later [1].

*Extended version was published in Proc. of the 9th Student Research Conference in Informatics and Information Technologies (IIT.SRC 2013), STU Bratislava, 257-262.*

*Acknowledgement.* This contribution is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

## References

- [1] Biswas, G., Leelawong, K.: Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*, (2005), pp. 363–392.
- [2] Hundhausen, Ch.D., Agarwal, P., and Trevisan, M. (2011): *Online vs. face-to-face pedagogical code reviews: an empirical comparison*. Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE'11), ACM, New York, NY, USA, pp. 117 – 122.