Source Code Review Recommendation

Matej CHLEBANA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xchlebana@stuba.sk

Quality assessment software is complex, but substantial activity in terms of the project success. Early detection of errors can alert management of the company on issues that may lead to prolonging or complete abolition of the project. Except for detecting errors it also helps maintaining consistency of source code and reducing the risks associated with the departure of one of the team members. Revision of developers' source code is time-consuming especially in larger companies where the new versions of the source code are generated in small intervals of time.

Software project management has a complex and extensively defined position. Project managers monitor and control the work of designers, developers and testers, but sometimes they also actively participate in these activities. While developers focus on the source code itself, architecture and performance, managers focus on a higher level, which includes: direction of the project, allocation of resources, functionality and user experience [1]. As the developers and managers focus on other activities in the project so they are interested in other data. Quality assessment takes into account various metrics. One group of metrics is to measure the activity of developers to analyze the mutual collaboration of developers in the team [2]. Conditions in which the source code originates attempts to capture systems for monitoring tasks. These systems capture the process of software development.

Evaluation of data obtained from monitoring of activities of the developers is a challenging process. Every developer is different, has different experiences and different strengths. Some developers create better source code during the day, another at night when they are not influenced by the noise of the surrounding area. Circumstances related directly to the developer have an impact on the quality of the source code, such as illness or different situations in life.

Identification of these special circumstances and characteristics of developers is difficult. Even some environmental influences on the work of the programmer cannot be detected (e.g., problems they are currently dealing with in their private lives). Although we are not able to fully identify these special circumstances, we can work

* Supervisor: Karol Rástočný, Institute of Informatics and Software Engineering

Spring 2014 PeWe Workshop, March 21, 2014, pp. 65-66.

with the information we collect, for example, through a system being developed within the research project PerConIK (Personalized Conveying of Information and Knowledge). PerConIK system aims to support business applications in software house using empirical software metrics. This system collects empirical data through software tools and extensions for development environments (Microsoft Visual Studio 2012 and Eclipse) and web browser (Mozilla Firefox) that are installed on the workstation of developers [3]. This data are, however, necessary to analyze and find metrics that are rated by different source code. Through process mining we will be able to create a model in which can identify and select potential risk source code so we try to make the best recommendation for checking the source code designated by a judge.

Motivation for examining this area is mainly to increase the success rate of software products by trying to warn the code reviewer for potential risk source code (whether due to a higher likelihood of errors or breaking of the coding conventions). The aim will be to separate "good" from potentially hazardous source code therefore to eliminate the necessity of reviewing all codes.

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] Buse, R. P. L., & Zimmermann, T. (2012). Information needs for software development analytics. In *2012 34th International Conference on Software Engineering (ICSE)* (pp. 987–996). Ieee. doi:10.1109/ICSE.2012.6227122

[2] Corral, L., Sillitti, A., Succi, G., Strumpflohner, J., & Vlasenko, J. (2012). DroidSense: A Mobile Tool to Analyze Software Development Processes by Measuring Team Proximity. In C. A. Furia & S. Nanz (Eds.), *Objects, Models, Components, Patterns* (Vol. 7304, pp. 17–33). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-30561-0_3

[3] Rástočný, K., & Bieliková, M. (2011). Information Tags as Information Space for Empirical and Code-based Software Metrics. In Proceedings of 17th International Conference on Fundamental Approaches to Software Engineering. (submitted)