Source Code Search Acknowledging Reputation of Developers

Martin MĚKOTA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
martin.mekota@gmail.com

Newcomers in big software development teams can be assigned to work on a difficult tasks right from the start. From a new member's perspective finding the right person to get an advice from may prove to be both time consuming and challenging since they might not be acquainted with other team members.

A prerequisite of an effective communication is the ability to identify a colleague to communicate with. The research of this area has helped to clarify, how much time do the developers spend communicating with colleagues. It is estimated that developers spend about 16% of their work time communicating. At the beginning of a software project this number rises to 50% [1].

Another study identified the key factors which people take into consideration when they are asking for help. People are afraid that they will look incompetent when they ask for help. They prefer private conversation when asking for help. They also want to spend as little time as possible looking for a person who can help them. In another words, they want to ask the right person on the first try. The last point will be focus of our work [2].

In our work we are attempting to solve this problem by gathering and analyzing information from version control and issue tracking systems and presenting reputable experts. However, we do not want to force developers to do any additional work like filling questionnaire or rating like in other similar work [3]. We want to assess their reputation and expertize by analyzing data which they produce in their current workflow. The end result of my thesis will recommend these experts in certain parts of the source code therefore new members will spend less time find them and more time discussing the problem. The reputation of the experts will be based on their activity in the issue tracking system and in version control system where code reviews take place.

At the first we used open source code repositories of Eclipse foundation to collect data about the developers. We used Node.js based web scraper to scrape relevant information from all the commits pushed to the repository since the project's beginning.

[*] Supervisor: Eduard Kuric, Institute of Informatics, Information Systems and Software Engineering

Spring 2016 PeWe Workshop, April 2, 2016, pp. 13–14.

The main information we are interested in is who made the commit, which files does the commit contain and link to the Gerrit Code Review. We will use all of these data to assess the reputation of a developer as well as show his activity in the project. From the Gerrit Code Review, we can extract information about, who was the code reviewer, what rating did he give to a change request and how many patches needed to be submitted to accept the change. From the data from the Gerrit Code Review tool, we will try to access reputation of members of the team. All of these data is produced by developers in their day to day work, no extra effort on the developers' part is required. This is the key idea behind our work.

To present the information from analysing the data we will implement a plugin to the Eclipse IDE showing our results. A person will be able to search the source files using Hound source code search engine. The results of a Hound search will be display in the plugin in addition to the information about who has been making changes to individual file and their reputation.

References

[1] Mockus A, Herbsleb J, „Expertise Browser: a quantitative approach to identifying expertise,“ , pp. 503-512, 2002.

[2] Ye Y, Yamamoto Y, Nakakoji K, „A socio-technical framework for supporting programmers,“ Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering, pp. 351-360, 2007.

[3] D. W. McDonald a M. Ackerman, „Expertise recommender,“ Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00, pp. 231-240, 2000.