Modelling Developer's Expertise

Eduard KURIC*

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

Estimation of developer's expertise allows, e.g., developers to locate an expert in a particular library or a part of a software system (someone who knows a component or an application interface). In a software company estimation of developers' expertise allows, for example, managers and team leaders to look for specialists with desired abilities, form working teams or compare candidates for certain positions. It can be also used to support so-called "search-driven development". Relevance of software artifacts is of course paramount, however, trustability is just as important. When a developer reuses a software artifact from an external source he has to trust the work of an external developer who is unknown to him. If a target developer would easily see that a developer with a good level of expertise has participated in writing the software artifact, then the target developer will be more likely to think about reusing.

Our new idea is to establish automatically developer's karma based on monitoring his working activities during coding in integrated development environment (IDE), analyzing and evaluating the (resultant) code he creates and commits to local repository. Primarily, we focus on answering these questions:

- What is the overall developer's karma and contribution for a software project, i.e., a degree (level) of his experience and familiarization with a functionality of a software system?
- Who has appropriate expertise for a particular part (component) of a software project?

In this area, several approaches have been proposed. The Expertise Browser [3] and Expertise Recommender [1] use a simple heuristic (Line 10 Rule) that the developer who commits to a file has expertise in the file. The Expertise Recommender establishes developer's expertise as a binary function, i.e., at a certain time only one developer can have expertise in a file and it depends on who last changed it (last one wins). An argument is that the developer who last made a change has the code "freshest" in mind. It has limitations because it does not take into account a degree of real developer's source code contributions, i.e., it does not reflect his overall know-how about the all

* Supervisor: Mária Bieliková, Institute of Informatics and Software Engineering

Spring 2014 PeWe Workshop, March 21, 2014, pp. 71–72.

code. The Emergent Expertise Locator [2] and Expertise Recommender improve the Expertise Browser so that they consider a relationship among changes in a file during determining expertise. However, the both approaches represent developer's expertise too straightforward, i.e., a developer who completely changed the implementation of a code fragment has no influence on the expertise of the developer who created the code.

On the contrary of existing approaches to establishing developer's expertise, we analyze his developing activities and the (resultant) code more deeply. To establish the overall developer's karma for a software project, we investigate software artifacts (components) which the developer creates. In other words, we take into account developer's "karma elements" as:

- *degree of authorship* - the developer's code contributions and the way how the contributions were created to the component;
- *authorship duration and stability* - the developer's know-how persistency about a component;
- *technological know-how* - the level of how the developer knows the used technologies (libraries), i.e., broadly/effectively;
- *component importance*.

We have developed a tool for monitoring how the developers create code. It allows us to capture, track and evaluate different events, e.g., we log copy/paste, find and navigation actions in a web browser, Visual Studio, OneNote; biometric information; utilization of HW resources; and states of running apps. We have inferred the particular developer's karma elements based on our observation of developers' activities.

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] McDonald, D.W., Ackerman, M.S.: Expertise recommender: a flexible recommendation system and architecture. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work* (CSCW '00). ACM, New York, USA, 2000, pp. 231-240.

[2] Minto, S., Murphy G.C.: Recommending Emergent Teams. In *Proceedings of the Fourth International Workshop on Mining Software Repositories* (MSR '07). IEEE Computer Society, Washington, USA, 2007, pp. 5-.

[3] Mockus, A., Herbsleb, J., D.: Expertise browser: a quantitative approach to identifying expertise. In *Proceedings of the 24th International Conference on Software Engineering* (ICSE '02). ACM, New York, USA, 2002, pp. 503-512.