

# Automatic Estimation of Software 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*

Expert recommendation systems in software engineering help to locate (discover) and to recommend individuals (experts) who have appropriate expertise on a given source code artefact [2,3,4]. Estimation of developer's expertise can be a valuable asset for a software company. It can be beneficial in the planning of a software project, especially in assigning development tasks. The time required to implement a new functionality, to change an existing functionality, or to fix a bug can be significantly reduced if the (issue) task is assigned to a developer who knows corresponding source code.

Despite different automatic expertise metrics were proposed we are not able to determine which metrics most reliably reflect developer's expertise. The crucial problem in comparing expertise metrics is the lack of a clear baseline with which to compare the metrics to each other. Another problem is that there are many competing definitions of expertise in literature. Many determinants may influence developer's expertise, e.g., knowledge of programming language and technologies, abilities of applying design patterns, a level of testing, and familiarity with documentation. Different approaches are required for establishing specific knowledge and skills.

Developer's expertise can be defined as a degree of his/her familiarity with a software system (software project), respective to other developers of the system [1]. Existing approaches to estimate developer's expertise on a part of a software system rely on the assumption that a number of lines of code committed by a developer reflects his/her expertise on that part of the system. However, we believe that in addition to the amount of final code we should also consider how much effort he/she put into implementation of the code.

In our work we propose an automatic approach to identify and to recommend an expert for a given development task of a software project that considers both a degree of developer's familiarity with the task and its corresponding source code, and his/her development productivity.

---

\* Supervisor: Mária Bielíková, Institute of Informatics, Information Systems and Software Engineering

A software system can be viewed as a body of knowledge decomposed into a set of fragments called conceptual concerns (topics). We estimate developer's familiarity with the software system at level of topics. We build a topic model to extract topics from codebase of the system. A degree of developer's familiarity with a topic is estimated from his/her code contributions to source code of the topic.

We estimate developer's productivity as complexity (size) of code changes performed by the developer per time and the amount of effort he/she spent to perform the changes measured through his/her development activities.

We also propose an approach that maps tasks in natural language to topics inferred from source code. To recommend an expert for a topic we combine developer's familiarity with this topic and his/her development productivity.

We evaluate our approach in three environments - an open environment (open-source projects), a commercial / closed environment, and an academic environment. The results indicate that by using our approach we are able to recommend developers who are competent to participate in and contribute to resolving newly created tasks. By recommending the right person to a given task we can reduce the overall time needed to resolve the task of the extra time that a developer would need to become familiar with the task and its corresponding code.

## References

- [1] Fritz, T., Murphy, G. C., Murphy-Hill, E., Ou, J. and Hill, E. (2014). Degree-of-knowledge: Modeling a developer's knowledge of code. *ACM Trans. Software Eng. Methodol.*, 23 (2), pp. 14:1-42.
- [2] Minto, S. and Murphy, G. C. (2007). Recommending emergent teams. In *Proceedings of the 4th International Workshop on Mining Software Repositories*, IEEE CS, pp. 5-12.
- [3] Mockus, A. and Herbsleb, J. D. (2002). Expertise browser: a quantitative approach to identifying expertise. In *Proceedings of the 24th International Conference on Software Eng.*, ACM, pp. 503–512.
- [4] Robillard, M., Maalej, W., Walker, R. and Zimmermann, T. (2014). *Recommendation Systems in Software Engineering*. Springer Berlin Heidelberg.