Automatic Estimation of Developer's Expertise

Eduard KURIC*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 841 04 Bratislava, Slovakia
eduard.kuric@stuba.sk

Evaluating expertise of developers is critical in software engineering, in particular for effective code reuse [2]. In a software company, technical and expert knowledge of employees is usually not represented in a unified manner and it is difficult to measure or observe directly. The level of developer's expertise is problematic to determine/estimate automatically. For example, in order for the developer to demonstrate expertise with a particular technology (e.g. software library) we need to administer a test. However, there is often a problem to motivate developers to participate in a test, and additional different developers often have different standards for judging the degree of expertise. Therefore, our steps are based on the automatic estimation of the relative expertise with the consideration of other developers and to compare them with each other in the company.

Our new idea is to estimate developer's expertise automatically based on monitoring his or her working activities during coding in integrated development environment (IDE), analyzing and evaluating the (resultant) source code he or she creates and commits to local repository. By applying our approach we are able to observe and evaluate different indicators. For example, we can distinguish a developer who often copies and pastes source code from an external source (Web). Source code contributions of such developer can be relative to the software project, moreover, it can reveal a reason of his frequent mistakes or low productivity in comparison with other developers.

The heart of our approach to developer's expertise modelling is a structured *domain model*. A software system can be viewed *as a body of knowledge* decomposed into a set of fragments called *conceptual concerns*. In software domain *conceptual concerns* of a software system refer to main technical concepts that reflect business logic or domain of the system [1].

Our developer's model overlays a domain model of the software system. It stores information for estimating his or her expertise related to particular concerns. In other words, we say that developer's familiarity (expertise) with a software system or its part

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

Spring 2015 PeWe Workshop, April 11, 2015, pp. 111–112.

is related, layer to layer, to the concerns of the software system (part), producing the developer's expertise model.

The developer's model is gradually modified when the overlay model receives information on developer's interaction captured in an *interaction model*. The interaction model represents and defines the interactions between a developer and concerns at a level of source code entities. The data stored in the interaction model are used to infer developer's characteristics. Concerns can be automatically extracted using statistical topic modelling techniques adapted to software. Such techniques can be used to approximate software concerns as (latent) topics.

The domain model of a software system consists of *metadata layer* and *topics layer*. It represents conceptualization of resources that are modified, created and (re)solved by developers. The proposed developer's model overlays the domain model. That is, developer's characteristics are related to domain model elements that are in our case metadata elements and topics, respectively. The characteristics are gathered at two levels, namely, developer's *personal and inferred characteristics*, and *relationships characteristics*.

The developer’s model is gradually modified through an *interaction model*. Relationships between the developer and the domain model elements are created based on developer's interactions with corresponding resource elements. Data stored in the interaction model are used to infer developer’s relationships characteristics. Separate developer's model is instantiated for each particular developer of a software system. The model is used to estimate developer’s expertise at the level of topics [3, 4].

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] Chen T-H., et. al.: Explaining software defects using topic models. In: *Proc. of the 9th IEEE Working Conference on Mining Software Repositories* (MSR '12). IEEE Press, USA, 2012, pp. 189-198.

[2] Gallardo-Valencia, R., E., et al.: Searching for reputable source code on the web. In: *Proc. of the 16th ACM international conference on Supporting group work* (GROUP '10). ACM, USA, 2010, pp. 183-186.

[3] Kuric, E., Bieliková, M.: Webification of Software Development: User Feedback for Developer's Modeling. In: *Proc. of the 14th International Conference on Web Engineering* ICWE’14. Springer LNCS, France, 2014, pp. 550-553.

[4] Kuric, E., Bieliková, M.: Estimation of student's programming expertise. In: *Proc. of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (ESEM '14). ACM, USA, 2014, 4 p.