Empir Software Eng (2016) 21:301–302
DOI 10.1007/s10664-016-9428-6

EDITORIAL

Guest editorial: Special section on mining software repositories

Massimiliano Di Penta[1] · Sunghun Kim[2]

Published online: 12 March 2016

The goal of mining software repositories is to retrieve, integrate and analyze data originating from different kinds of software repositories, such as versioning systems, issue trackers, mailing lists, or forges. The purpose of such analysis could be, on the one hand, of observing the evolution of software projects, and on the other hand of building different forms of recommenders to support software engineering in their activities.

This special section features three mining software repository papers dealing with one very important aspect of software evolution, i.e., software defects, and specifically their prediction, localization, and triaging.

In the paper *"The Impact of Tangled Code Changes on Bug Defect Prediction Models"* (Herzig et al. 2015) study the impact of tangled changes on defect prediction models. Tangled changes occur when developers commit together unrelated code changes within a single transaction. The authors found, on five Java open source projects, that such commits impact between 7 % and 20 % of bug fixes, making at least 16 % of files incorrectly associated to bug reports. Untangling code changes would result in a statistically significant improvement of defect prediction accuracy, between 5 % and 200 %.

In the paper *"Improving Bug Management using Correlations in Crash Reports"* (Wang et al. 2014) use crash correlation groups, i.e., groups of crash types related to identical or related bug reports, to perform bug localization as well as to identify duplicate bug reports. Results of a study conducted on data from Eclipse and Firefox show the applicability of the proposed approach, i.e., when identifying the top three defect-prone file candidates, the approach exhibits recall between 52 % and 62 % and a precision between 42 % and 50 %.

✉ Massimiliano Di Penta
dipenta@unisannio.it

Sunghun Kim
hunkim@cse.ust.hk

[1] Department of Engineering, University of Sannio, Benevento, Italy

[2] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

The identification of duplicate bug reports exhibits a recall between 47 % and 50 %, and a precision between 35 % and 55 %.

The paper *"A Contextual Approach towards More Accurate Duplicate Bug Report Detection and Ranking"* by Hindle et al. (2015) also deals with duplicate bug reports. In particular, the authors investigate how contextual information related to software quality attributes, terms related to software architectures, and system-development topics can be used to improve bug deduplication. Results of a study conducted on Android, Eclipse, Mozilla, and OpenOffice show that contextual information should not be ignored when using information retrieval tools for bug report deduplication.

References

Herzig K, Just S, Zeller A (2015) The impact of tangled code changes on defect prediction models. Empirical Software Engineering, 1–34, doi:10.1007/s10664-015-9376-6

Hindle A, Alipour A, Stroulia E (2015) A contextual approach towards more accurate duplicate bug report detection and ranking. Empirical Software Engineering, 1–43, doi:10.1007/s10664-015-9387-3

Wang S, Khomh F, Zou Y (2014) Improving bug management using correlations in crash reports. Empirical Software Engineering, 1–31, doi:10.1007/s10664-014-9333-9