Guest Editorial: Mining software repositories

Romain Robbes¹ · Yasutaka Kamei² · Martin Pinzger³

Published online: 26 May 2017

© Springer Science+Business Media New York 2017

The Mining Software Repositories (MSR) field analyzes the rich data available in software repositories to uncover interesting and actionable information about software systems and projects. Some commonly explored areas include software evolution, models of software development processes, characterization of developers and their activities, prediction of future software qualities, use of machine learning techniques on software project data, software bug prediction, analysis of software change patterns, and analysis of code clones. This special issue provides five recent MSR regular research papers, and, for the first time in the Journal of Empirical Software Engineering, three data showcase papers. Each of these data showcase papers describe at length a valuable Software Engineering dataset, in the hope that it allows prospective users of these datasets a smooth start with them. In the following, we first introduce the five regular research papers and then the three data showcase papers.

The paper “[A Large-Scale Study of Architectural Evolution in Open-Source Software Systems](#)” by Behnamghader, Le, Garcia, Link, Shahbazian, and Medvidovic introduces ARCADE, an architecture recovery framework for conducting large-scale replicable empirical studies of architectural changes across different versions of a software system. Using ARCADE on 23 open-source systems, the authors report several findings that corroborate a number of widely held views about the times, frequency, scope, and nature of architectural changes.

✉ Romain Robbes
rrobbes@unibz.it

Yasutaka Kamei
kamei@ait.kyushu-u.ac.jp

Martin Pinzger
martin.pinzger@aau.at

¹ Software and Systems Engineering Research Group (SwSE), Free University of Bozen-Bolzano, Bolzano, Italy

² Principles of Software Languages (POSL) Group, Kyushu University, Fukuoka, Kyushu, Japan

³ Software Engineering Research Group, Alpen-Adria Universität Klagenfurt, Klagenfurt, Austria

The paper “[Analysis of License Inconsistencies in Large Collections of Open Source Projects](#)” by Wu, Manabe, Kanda, and German categorizes different types of license inconsistencies and presents a method to detect them. Using the method, the authors detect license inconsistencies in Debian 7.5 and a collection of more than 10k Java open source projects. Some of them are manually analyzed and classified into four categories. The results clearly show that license inconsistencies exist and that two categories of license inconsistencies indicate license problems that require the developers’ attention.

In the paper “[Predicting the Delay of Issues with Due Dates in Software Projects](#)”, the authors Choetkiertikul, Dam, Tran, and Ghose present an approach to support project managers in predicting whether an issue is at risk of being delayed and the extent of the delay. 19 risk factors extracted from the issues of eight open source projects are used to train the prediction models. The evaluation of the models shows that the likelihood of delay can be predicted with 9% precision and 61% recall. The extent of the delay can be predicted with a macro-averaged mean cost-error of 0.66 and macro-averaged mean absolute error of 0.72.

The paper “[Exception Handling Bug Hazards in Android: Results from a Mining Study and an Exploratory Survey](#)” by Coelho, Almeida, Gousios, van Deursen, and Treude first presents a detailed empirical study of common bug hazards in over 6,000 Java exception stack traces extracted from over 600 open source Android projects. The bug hazards found are further assessed in a survey with 71 developers involved in at least one of the projects analyzed. The findings call for tool support to help developers understand their own and third party exception handling and wrapping logic.

In the paper “[Do Bugs Foreshadow Vulnerabilities?: An In-depth Study of the Chromium Project](#)”, the authors Munaiah, Camilo, Wigham, Meneely, and Nagappan examine the relationship between bugs and vulnerabilities. The authors mined 374,686 bugs and 703 post-release vulnerabilities over five Chromium releases that span six years of development. The results indicate that bugs and vulnerabilities are empirically dissimilar groups, motivating the need for security engineering research to target vulnerabilities specifically.

The data showcase paper “[Fine-GRAPE: Fine-Grained API Usage Extractor—an Approach and Dataset to Investigate API Usage](#)” by Sawant and Bacchelli presents an approach to extract type-checked API method invocation information from Java programs. Using the method, the authors mine a total of 20,263 projects including 1,482,726 method invocations and 85,098 annotation usages related to 5 popular and established APIs (Spring, Hibernate, Guava, Guice, and EasyMock). The dataset is available online as a PostgreSQL data dump on FigShare.

In the data showcase paper “[A Repository of Unix History and Evolution](#)”, the author Spinellis aims at preserving the UNIX development history and offering it to the software engineering community. He creates a repository as a Git version control system providing the data of 24 snapshots of the UNIX system developed at Bell Labs, the University of California at Berkeley, and the 386BSD team, two legacy repositories, and the modern repository of the open source FreeBSD system. The repository documents the detailed history and evolution of a foundational software system over a period of 44 years.

Finally, the data showcase paper “[The Debsources Dataset: two Decades of Free and Open Source Software](#)” by Caneill, Germañ, and Zacchiroli presents source code and related metadata over a period of two decades of Debian distributions. The dataset includes 3 billion lines of source code, corresponding to 10 Debian stable releases, as well as metadata related to them such as size metrics (lines of code, disk usage) and license information (GPL, BSC,

etc). The dataset is available online as a set of tarballs and a PostgreSQL data dump, hosted on Zenodo.

Acknowledgments We are grateful for the continuous support and encouragement offered by the Editorial Board and the Editors-in-Chief Robert Feldt, Thomas Zimmermann, and Lionel Briand. We also thank the authors for keeping up with the review schedule and the reviewers for their detailed and constructive comments which helped to shape the papers.