Guest editorial: Mining software repositories 2018

Yasutaka Kamei[1] · Andy Zaidman[2]

During software development software engineers use a variety of tools and platforms to create, analyze, validate, and collaborate on software systems. These tools and platforms contain rich data that has the potential to bring deep insight into both the software systems themselves, and the underlying engineering processes. The Mining Software Repositories (MSR) community is actively working on tools and techniques that analyze these rich data sets.

Thanks to readily available version control systems, issue trackers, mailing lists, build logs, and other sources of information from open source projects, repository mining has gained in popularity since the inception of the Mining Software Repositories series in 2004. It continues to be one of the fastest growing fields in the area of software engineering. Researchers in this field empirically explore a range of software engineering questions using software repository data as the primary source of information. Some commonly explored areas include software evolution, models of software development processes, characterization of developers and their activities, use of machine learning techniques on software project data, software bug prediction, analysis of software change patterns, analysis of code clones, mining code review data, and mining execution traces and logs.

This special section on Mining Software Repositories (MSR) serves to highlight and elaborate on papers from the 2018 edition of the International Conference on Mining Software Repositories. In particular, it features two recent MSR full research papers that touch upon important aspects; one on software evolution and the other on software analytics. The two full research papers were invited and extended with a significant amount of new and substantive materials for journal publication. Each paper was reviewed by three or more reviewers and the authors faced (at least) one major revision. In the following, we briefly discuss the two full research papers.

The paper "CDA: Characterising Deprecated Android APIs" by Li, Gao, Bissyandé, Ma, Xia, and Klein introduces CDA, a research-based prototype tool for characterizing deprecated Android

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

Andy Zaidman
A.E.Zaidman@tudelft.nl

[1] Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan

[2] Software Engineering Research Group, Delft University of Technology, Delft, The Netherlands

APIs. The authors apply CDA to different revisions (e.g., releases) of the Android framework code. They investigate six research questions (e.g., Are deprecated APIs properly annotated and documented in the Android framework code base?) using a dataset that comprises 3 million lines of Java code. The results indicate that the proposed approach has identified three bugs related to deprecated APIs. The authors make their implementation available online at https://github.com/lilicoding/CDA.

Agrawal, Menzies, Minku, Wagner, and Yu are the authors of the paper "Better Software Analytics via "DUO": Data Mining Algorithms Using/Used-by Optimizers". In this paper they provide their vision on empirical software engineering research and practice, with a particular focus on the complementarity of the optimization of data-mining (largely the focus of the software analytics/MSR community) and the optimization of parameters (largely the focus of the search-based software engineering community). The authors call this combination DUO, short for data miners using/used by optimizers. The authors make four claims (e.g., for software engineering tasks, optimizers can greatly improve data miners) and support them based on a literature review of applications of DUO.

We hope that you gain much from this special issue on Mining Software Repositories. Enjoy!

Acknowledgments We are grateful for the continuous support and encouragement offered by the editorial board of the Journal of Empirical Software Engineering and by the Editors-in-Chief Robert Feldt and Thomas Zimmermann. 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.

Yasutaka Kamei is an associate professor at Kyushu University in Japan. He has been a research fellow of the JSPS (PD) from July 2009 to March 2010. From April 2010 to March 2011, he was a postdoctoral fellow at Queen's University in Canada. He received his B.E. degree in Informatics from Kansai University, and the M.E. degree and Ph.D. degree in Information Science from Nara Institute of Science and Technology. His research interests include empirical software engineering, open source software engineering and Mining Software Repositories (MSR). He served as a program-committee co-chair of SANER 2016 and MSR 2018. He is an editor of the Springer Empirical Software Engineering Journal (2016-present). More information about him is available online at https://posl.ait.kyushu-u.ac.jp/~kamei/.

Andy Zaidman is a full professor in software engineering at Delft University of Technology, the Netherlands. He obtained his MSc (2002) and PhD (2006) in Computer Science from the University of Antwerp, Belgium. His main research interests are software evolution, program comprehension, mining software repositories and software testing. He is an active member of the research community and involved in the organization of numerous conferences (WCRE’08, WCRE’09, VISSOFT’14 and MSR’18). He is on the editorial board of JSS and EMSE. In 2013 he was the laureate of a Vidi career grant from the Dutch science foundation NWO, while in 2019 he won the Vici career grant, the most prestigious career grant from the Dutch science foundation NWO.