Empir Software Eng (2012) 17:500–502
DOI 10.1007/s10664-012-9206-z

EDITORIAL

Introduction to the Special Issue on Mining Software Repositories in 2010

Jim Whitehead · Thomas Zimmermann

Published online: 26 April 2012

This special issue of Empirical Software Engineering consists of revised and extended versions of three selected papers originally presented at the *7th IEEE Working Conference on Mining Software Repositories* (MSR 2010). The conference was held in Cape Town, South Africa, on May 2–3, and was co-located with the *32nd ACM/IEEE International Conference on Software Engineering* (ICSE 2010). This conference brings together researchers who share an interest in advancing the science and practice of software engineering via the analysis of data stored in software repositories.

The Mining Software Repositories field analyzes the rich data available in software repositories to uncover interesting and actionable information about software systems and projects. Thanks to the ready availability of software configuration management, mailing list, and bug tracking repositories from open source projects, it has gained popularity since 2004 with the first instance of the MSR workshop (now conference) and 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, 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. There has also been a stream of work on tools for mining software repositories, and techniques for visualizing software repository data.

In recent years the importance of the field has further increased as today's society and businesses become more data-driven. Industry has a strong interest in transforming software repository data into actionable insights to inform better development decisions. Analytics is already commonly used in many businesses—notably in marketing, to better reach and understand customers (Thornton 2009). The application of analytics to software development is becoming more popular.

J. Whitehead
University of California, Santa Cruz, Santa Cruz, CA 95064, USA
e-mail: ejw@soe.ucsc.edu

T. Zimmermann (✉)
Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
e-mail: tzimmer@microsoft.com

For those wishing to develop a broad understanding of the software repository mining research field, there are several resources.

- Tao Xie and Ahmed E. Hassan provide an overview of software repository mining research areas and methods in the tutorial notes to their Mining Software Engineering Data tutorial, given at several recent software engineering conferences (notes are available at http://ase.csc.ncsu.edu/dmse/).
- Kagdi et al. provide a survey of software repository mining techniques in (Kagdi et al. 2007) that focuses on software evolution research, but which provides broad coverage of a wide range of research methods and repository types used in the field.
- Hassan provides a survey of the entire field along with future research challenges in (Hassan 2008). Together, these three sources give a strong general introduction to software repository mining research.
- The PROMISE repository (http://promisedata.org/) is a collection of 140+ software engineering data sets related to defect prediction, effort estimation, model-based software engineering and many other topics.

Papers in the Mining Software Repositories field tend to take a quantitative empirical approach to exploring research questions. As a consequence, it is natural to select the best papers from the MSR conference for inclusion in Empirical Software Engineering. The three papers in this special issue provide a good cross section of the topics and approaches recently explored in the mining software repositories community.

In the first paper, *"Clones: What is that Smell?",* Rahman, Bird, and Devanbu try to validate conventional wisdom that cloning makes code more defect-prone by analyzing the software repositories of four open-source projects. Assessing the validity of common software engineering folklore is a frequent application of mining software repositories. The findings in the paper do not support the claim that clones are generally a "bad smell"—especially with respect to defects. They found that clones may be even less defect-prone than non-cloned code. They also found little evidence that clones with more copies are actually more error prone. As put it in the paper, "perhaps we can clone, and breathe easily, at the same time."

In the paper *"Evaluating Defect Prediction Approaches: A Benchmark and an Extensive Comparison",* D'Ambros, Lanza, and Robbes introduce several novel datasets for defect prediction. As they put it "predicting software defects is one of the holy grails of software engineering". Over the past years, researchers have devised and implemented literally hundreds of defect prediction approaches (read the systematic review by Hall et al. (2011) for a good summary). However, the absence of benchmarks made it difficult to compare approaches. In their paper, D'Ambros et al. present a benchmark and provide an extensive comparison of well-known defect prediction approaches, together with novel approaches that they devised. The benchmark is available at http://bug.inf.usi.ch/

In the paper, *"The Evolution of Java Build Systems",* McIntosh, Adams, and Hassan study the build systems of six open-source projects. While build systems are important to create the executable files of software, especially in industry, build systems have largely been ignored by research until recently. McIntosh et al. observed that the sizes of the build system and source code are highly correlated and that often restructuring the source code also required restructuring the build system. Understanding build processes helps project managers to better allocate personnel and resources to the build system.

We hope you enjoy the papers in this special issue.

Acknowledgments We are grateful to the continuous support and encouragement offered by the Editorial board for the Journal of Empirical Software Engineering and by the Editor-in-Chief Lionel Briand. This issue

is the result of a great deal of effort by the reviewers, authors, and attendees of MSR 2010. We thank the authors for keeping up with the review schedule and the reviewers for their detailed and constructive comments which helped shape the papers.

References

Hall T, Beecham S, Bowes D, Gray D, Counsell S (2011) "A systematic review of fault prediction performance in software engineering," preprint, to appear in IEEE Transactions on Software Engineering. http://doi.ieeecomputersociety.org/10.1109/TSE.2011.103

Hassan A (2008) "The road ahead for mining software repositories." In: Frontiers of software maintenance, held with the 2008 IEEE International Conference on Software Maintenance, Beijing, China, pp. 48–57

Kagdi HH, Collard ML, Maletic JI (2007) A survey and taxonomy of approaches for mining software repositories in the context of software evolution. J Softw Maint 19(2):77–131

Thornton May (2009) The new know: innovation powered by analytics. Wiley

Jim Whitehead is a professor of Computer Science at the University of California, Santa Cruz, where he directs the Software Introspection Laboratory. His research interests in software engineering focus on software bug prediction, understanding the nature of bugs, software evolution, and software design. He received his PhD in Information and Computer Science from the University of California, Irvine in 2000.

Thomas Zimmermann is a researcher in the Research in Software Engineering Group at Microsoft Research, adjunct assistant professor at the University of Calgary, and affiliate faculty at University of Washington. His research interests include empirical software engineering, mining software repositories, software reliability, development tools, and social networks, and computer games. He received his PhD degree from Saarland University, Germany in 2008.