Empir Software Eng (2017) 22:963–966
DOI 10.1007/s10664-017-9509-1

EDITORIAL

Empirical studies in software and systems traceability

Patrick Mäder[1] · **Rocco Olivetto**[2] · **Andrian Marcus**[3]

Published online: 28 March 2017

Software and systems traceability is a critical element of any rigorous software development process and it is a required component of the approval and certification process in most safety- and security-critical systems. With the growing importance of these systems in our societies, traceability became a heavily studied research topic. Though important, traceability is also an element of software development processes that is most elusive. Manufacturers struggle in finding the right degree of traceability for their needs and in establishing accurate sets of traceability links. The cost, effort, and discipline needed to create and maintain trace links in a rapidly evolving software system can be extremely high. Moreover, its benefits often go unrealized in practice, either due to ill-defined and ad-hoc traceability processes, poor user training, or a lack of effective tool support. However, at the same time, traceability has been demonstrated as being able to reduce development effort and to improve development quality.

Traceability research is often based on rigorous empirical studies to explore new research questions or to evaluate new tracing approaches. In this special issue, we include papers that employ empirical techniques to evaluate four traceability link recovery approaches, based on information retrieval and natural language processing techniques, eye movements, and machine learning, and an approach for real-time traceability.

The first paper "*Achieving Traceability in Large Scale Continuous Integration and Delivery: Deployment, Usage and Validation of the Eiffel Framework*" [DOI: 10.1007/s10664-016-9457-1] presents, investigates and discusses Eiffel, an industry developed solution designed to provide real time traceability in continuous integration and delivery. The authors also investigate traceability needs of industry professionals through interviews, providing context to their solution. The method is found to address the identified traceability needs and found in some cases to reduce traceability data acquisition times from days to minutes, while at the same time alternatives offering comparable functionality are lacking. Further, traceability is shown not only to be an important concern to engineers, but also regarded as a prerequisite to successful large scale continuous integration and delivery.

✉ Patrick Mäder
patrick.maeder@tu-ilmenau.de

[1] Technische Universität Ilmenau, Ilmenau, Germany

[2] University of Molise, Campobasso, Italy

[3] The University of Texas at Dallas, Richardson, TX, USA

The second paper "*Estimating the Number of Remaining Links in Traceability Recovery*" [DOI: 10.1007/s10664-016-9460-6] introduces an approach called ENRL (Estimation of the Number of Remaining Links), which aims at estimating the number of remaining positive links in a ranked list of candidate traceability links produced by a recovery approach based on Natural Language Processing techniques. The authors evaluated the accuracy of their approach by considering several machine learning classifiers and NLP techniques on three datasets, and concerning traceability links among different kinds of software artifacts including requirements, use cases, design documents, source code, and test cases. Results indicate that: (i) specific estimation models are able to provide accurate estimates of the number of remaining positive links; (ii) the estimation accuracy depends on the choice of the NLP technique, and (iii) univariate estimation models outperform multivariate ones.

The third paper "*Automated Training-Set Creation for Software Architecture Traceability Problem*" [DOI: 10.1007/s10664-016-9476-y] presents three baseline approaches for the creation of training data for trace retrieval techniques based on machine learning. These approaches are (i) Manual Expert-Based, (ii) Automated Web-Mining, which generates training sets by automatically mining tactic's APIs from technical programming websites, and lastly (iii) Automated Big-Data Analysis, which mines ultra-large scale code repositories to generate training sets. The authors compare the trace-link creation accuracy achieved using each of these three baseline approaches and discuss the costs and benefits associated with them. Additionally, in a separate study, they investigate the impact of training set size on the accuracy of recovering trace links. Results indicate that automated techniques can create a reliable training set for the problem of tracing architectural tactics.

The forth paper "*Eye Movements in Software Traceability Link Recovery*" [DOI: 10.1007/s10664-016-9486-9] paper presents a comparative study on IR and eye-gaze based approaches performed on bug localizations task on the JabRef project. The authors present an algorithm that uses the collected gaze dataset to produce candidate traceability links related to the tasks. They found that the eye gaze approach outperformed standard Latent Semantic Indexing (LSI) and Vector Space Model (VSM) approaches when compared to how developers actually fixed the bug. Further, results show that links generated by their algorithm were more useful to fix the bug than those of the Information Retrieval (IR) technique for a majority of the tasks. Eventually, the authors also propose that the use of eye tracking could form an effective benchmark for bug localization techniques, complementary to the commit data currently used.

The fifth paper "*Tackling the Term-Mismatch Problem in Automated Trace Retrieval*" [DOI: 10.1007/s10664-016-9479-8] describes and compares three query augmentation techniques for addressing the term mismatch problem faced when applying information retrieval techniques to semi-automatically trace from regulatory codes to requirements. The first trains a classifier to replace the original query with terms learned from a training set of regulation-to-requirements trace links. The second, replaces the original query with terms learned through web-mining; and the third utilizes a manually constructed domain ontology to augment query terms. All three techniques were evaluated on security regulations traced to ten healthcare related requirements specifications. The classification approach returned the best results. The ontology-based solution showed smaller improvements, while the web-mining technique showed improvements in only a subset of queries. Overall, the techniques offer tradeoffs in terms of performance, cost, and usage viability within specific contexts.

We hope you enjoy the papers in this special issue. We also hope that such papers will help you to acquire knowledge on this field and shed some lights on new research directions.

Acknowledgements We are grateful for 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 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.

Patrick Mäder is a Professor at the Technische Universität Ilmenau, Germany and heading the endowed chair on Software Engineering for Critical Systems. His research interests include software engineering with a focus on requirements traceability, requirements engineering, object-oriented analysis and design, and development of safety-critical systems. Mäder received a Diploma degree in industrial engineering and a PhD degree (Distinction) in computer science from the Technische Universität Ilmenau in 2003 and 2009, respectively. He worked as a consultant for the EXTESSY AG, Wolfsburg, as postdoctoral fellow at the Institute for Systems Engineering and Automation (SEA) of the Johannes Kepler University Linz, Austria and as postdoctoral researcher at the Software and Requirements Engineering Center at the DePaul University Chicago, USA.

Rocco Olivetto is Associate Professor at University of Molise (Italy), where he is also the Chair of the Computer Science program and the Director of the Software and Knowledge Engineering (STAKE) Lab. He co-authored over 100 papers on topics related to software traceability, software maintenance and evolution, search-based software engineering, and empirical software engineering. His activities span various international software engineering research communities. He has served as organizing and program committee member of several international conferences in the field of software engineering. He was program co-chair of ICPC 2015, TEFSE 2015 and 2009, SCAM 2014, WCRE 2013 and 2012. He was also keynote speaker at MUD 2012.

Andrian Marcus is Associate Professor at The University of Texas at Dallas, in Richardson, TX, USA. His current research interests are focused on software evolution and program comprehension. He is best known for his more than decade-long work on using information retrieval and text mining techniques for software analysis to support comprehension tasks during software evolution, such as: concept location, impact analysis, error prediction, traceability link recovery, etc. He received several Best Paper Awards and two Most Influential Paper Awards, and he is the recipient of the NSF CAREER award in 2009. He was the Chair of the Steering Committee of ICSME and served on many conferences as chair and program committee member and also serves on the editorial board of three software engineering journals: IEEE Transactions on Software Engineering, Empirical Software Engineering Journal - Springer, and the Journal of Software: Evolution and Process - John Wiley and Sons.