Software Qual J (2018) 26:1415–1419
https://doi.org/10.1007/s11219-018-9410-1

Special issue on automation of software testing: improving practical applicability

Christof Budnik[1] · Gordon Fraser[2] ·
Francesca Lonetti[3] · Hong Zhu[4]

Published online: 19 May 2018

1 Test automation

An important challenge in software testing has been to improve the degree of attainable automation, in developing advanced techniques for generating the test inputs as well as in finding innovative support procedures to automate the testing process.

Test automation is a very active field of research, covering different aspects such as tools, application domains, the lifecycle development process, project dimensions, and empirical work reporting successful results or failures (Graham and Fewster 2012). Test automation is also highly relevant in practice: the market for automated test support tools is expanding, opening relevant business opportunities for new innovative testing platforms. The ultimate challenge is the development of a powerful integrated test environment that goes beyond automated test execution and by itself can automatically take care

✉ Christof Budnik
christof.budnik@siemens.com

Gordon Fraser
gordon.fraser@uni-passau.de

Francesca Lonetti
francesca.lonetti@isti.cnr.it

Hong Zhu
hzhu@brookes.ac.uk

[1] Siemens Corporation, Princeton, New Jersey, USA

[2] University of Passau, Passau, Germany

[3] ISTI-CNR, 56127 Pisa, Italy

[4] Oxford Brookes University, Oxford, UK

of selection, deployment, and integration of the testing tools that assist test activities across the software testing process (Garousi and Elberzhager 2017).

A high level of automation raises many challenges: The analysis of the large amount of test results produced automatically creates cost and effort. The trade-off between automated and manual testing has been frequently discussed in the literature (Garousi and Mäntylä 2016), and decision support systems for selecting the most effective and efficient testing tools for specific purposes in a specific context have been provided (Raulamo-Jurvanen 2017). Costs can be reduced by automating even test automation, for example, by using natural language test steps enabling a sequence of procedure calls with accompanying parameters that can drive testing without human intervention. This technique has been proven effective in reducing the cost of test automation by automating over 82% of the steps contained in a test suite (Thummalapenta et al. 2012). Finally, with a wider acceptance of test automation, the quality of test code or test scripts that perform test automation has become a major concern in practice and recently an active topic in research.

2 This special issue

This special issue focuses on a number of practical applicability aspects of test automation, including test suite performance, tester profile, and domain-specific language implementations as well as test model extraction and user interface testing, reflecting the frontier in research and the best practice in industry. It includes revised and extended versions of the best papers presented at the 11th IEEE/ACM International Workshop on Automation of Software Test (AST 2016), held in conjunction with the 38th International Conference on Software Engineering (ICSE'16), in Austin, TX, USA, May 14–22, 2016, as well as new original submissions.

This issue initially received a total of 13 submissions. Of these, 3 were withdrawn, and after a rigorous peer-review according to the journal's high standards, 4 papers have been rejected and 6 accepted.

This issue consists of the 6 papers that are briefly discussed below.

We have two papers looking at test prioritization, an approach which orders test cases to reduce the costs of finding faults. In "Similarity-Based Prioritization of Test Case Automation," Daniel Flemström, Pasqualina Potena, Daniel Sundmark, Wasif Afzal, and Markus Bohlin develop a prioritization technique that reuses already automated parts of test cases. In "Test Case Prioritization Techniques for Model-Based Testing: A Replicated Study," João Felipe Silva Ouriques, Emanuela Gadelha Cartaxo, and Patrícia Duarte Lima Machado present industrial case studies showing the factors influencing the performance of test case prioritization techniques in the context of model-based testing.

There are three papers looking at test automation in different domains: In "Automated Testing of DSL Implementations - Experiences from Building mbeddr," Daniel Ratiu, Markus Voelter, and Domenik Pavletic present their experience on testing different aspects of the implementation of domain-specific languages and associated tools, aiming at increasing the automation of language testing. In "Model extraction and test generation from JUnit test suites," Pablo Lamela Seijas, Simon John Thompson, and Miguel Ángel Francisco Fernández describe how to infer state machine models from legacy unit test suites and how to generate new tests from those models. In "Mobile GUI Testing," Inês

Coimbra Morgado and Ana C. R. Paiva present a tool for automating testing of mobile applications.

Finally, test quality is the topic of "An assessment of operational coverage as both an adequacy and a selection criterion for operational profile based testing" by Breno Miranda and Antonia Bertolino, who introduce the operational coverage concept that takes into account how much a program's entities are exercised in practice, in order to reflect the usage profile in the coverage measure.

Acknowledgments We thank the authors and reviewers of this special issue for their great effort and contributions. We also thank the *Software Quality Journal* production team for their hard work in assembling and editing the issue.

References

Garousi, V., & Elberzhager, F. (2017). Test automation: not just for test execution. *IEEE Software, 34*(2), 90–96.

Garousi, V., & Mäntylä, M. V. (2016). When and what to automate in software testing? A multi-vocal literature review. *Information and Software Technology, 76*, 92–117.

Graham, D. & Fewster, M. (2012) Experiences of test automation: case studies of software test automation. Addison-Wesley Professional.

Raulamo-Jurvanen, P. (2017). Decision support for selecting tools for software test automation. *ACM SIGSOFT Software Engineering Notes, 41*(6), 1–5.

Thummalapenta, S., Sinha, S., Singhania, N., & Chandra, S. (2012) Automating test automation. In 34th International Conference on Software Engineering (ICSE), pages 881–891.

Dr. Christof J. Budnik is a Senior Key Expert Engineer at the Architecture and Verification of Intelligent Systems Research Group of Siemens Corporation, Corporate Technology in Princeton, NJ. He leads research and business projects in several industrial domains and strives for advanced verification solutions using artificial intelligence and machine learning. Before joining Siemens he was the head of the software quality department for a German company in the smart card business responsible for the authentication, authorization and accounting of secure chip cards. Dr. Budnik obtained his Ph.D. in Electrical Engineering 2006 from the University of Paderborn, Germany, on event-based testing. He is author of many published contributions at several journals and conferences comprising his research interests in the areas of model-based testing, mutation testing, test automation, and formal verification. He organizes workshops on software testing, is PC member of many international conferences and serves as guest reviewer for selected journals.

Prof. Gordon Fraser is a Full Professor in Computer Science at the University of Passau. He received his Ph.D. from Graz University of Technology, Austria, in 2007, then worked as a post-doc researcher at Saarland University, Germany, and as a (Senior) Lecturer at the University of Sheffield until 2017. He has published on improving software quality and programmer productivity at all major software engineering venues (e.g., TSE, TOSEM, ICSE, ISSTA, FSE, ASE, ICST) and has received six ACM SIGSOFT Distinguished Paper Awards (FSE'14, ISSTA'14, ASE'14, ASE'15, ESEC/FSE'15, ICSE'17), as well as best paper awards at SSBSE and GECCO. He is chair of the steering committees of the International Conference on Software Testing, Verification, and Validation (ICST) and the steering committee of the International Symposium on Search-Based Software Engineering (SSBSE). He has been programme chair of software engineering conferences (ASE, ICST, TAP, TAIC PART, SSBSE) and workshops, is a regular member of many programme and organising committees in the field (e.g., ICSE, FSE, ASE, ISSTA), is associate editor of the IEEE Transactions on Software Engineering (TSE) and Software Testing, Verification, and Reliability (STVR) journals.

Dr. Francesca Lonetti is a researcher at CNR-ISTI, Italy. Her current research focuses on monitoring and testing of software systems. In particular, she is interested in: model-based monitoring approaches; testing of security systems (access control and usage control systems); methodologies and tools for robustness testing of web services, and cloud-based testing. Her expertise on these research topics has been applied in the context of several national and European research projects including LearnPAd, CHOReOS, TAS3, NESSoS, CONNECT, D-ASAP, ElasTest. She currently serves as member of the Editorial Board of International Journal of Communication Networks and Information Security (IJCNIS). She is and has been part of the program committee of several international conferences and workshops in the field, such as ICST, MODELSWARD, AST, QUATIC, ENASE. She has been publications chair of ICST 2012 and proceedings co-chair of ICSE 2015. She was co-chair of: the 11th IEEE/ACM International Workshop on Automation of Software Test (AST 2016) and International Workshop on domAin specific Model-based AppRoaches to vErificaTion and validaTiOn (AMARETTO 2016).

Dr. Hong Zhu is a professor of computer science at the Oxford Brookes University, Oxford, UK, where he chairs the Applied Formal Methods Research Group. He obtained his BSc, MSc and PhD degrees in Computer Science from Nanjing University, China, in 1982, 1984 and 1987, respectively. He worked at Nanjing University from 1987 to 1998 before he joined Oxford Brookes University in November 1998 as a Senior Lecturer in Computing. He became a Professor of Computer Science in October 2004. He is a senior member of IEEE, a member of British Computer Society, and ACM. His research interests are in the area of software development methodologies, foundation of software engineering, software design, modeling, software testing, Software-as-a-Service, etc. He has published 2 books and more than 190 research papers in journals and international conferences. He is the founder of AST workshops and Chair of the AST steering committee from 2006 to 2017.