DOI: 10.1002/smr.2510

EDITORIAL

Journal of Software: Evolution and Process WILEY

Guest editors' introduction to the special issue "Automatic Software Testing from the Trenches"

Software testing is an integral and important part of the software engineering discipline, and its automation has been actively pursued in both academia and industry to reduce its high costs. In the past decades, a considerable research effort has been devoted to automatic test case generation, automatic test selection, and automatic test oracles. The practice of software test automation has also moved forward significantly, and in recent years, a large number of software test tools have been developed and released to the market. However, progress in automatic software testing (AST) research is still required. The increasing complexity, pervasiveness and inter-connection of software systems, the ever-shrinking development cycles and time-to-market, and the scarcity of tools that can support all testing tasks within one environment, have posed new challenges and stricter constraints. Thus, despite significant achievements both in theory and practice, AST remains a challenging research area, and there is an urgent requirement to improve test automation to scale up productivity and quality in software development. Many times, however, industry needs differ from the research agenda, as companies need to prioritize reducing cost and time-to-market. Moreover, practitioners may have a hard time choosing a particular testing method or technology, since the real challenges that influence the decision are usually hidden.

This special issue includes revised and extended versions of the best papers presented at the 2nd ACM/IEEE International Conference on Automation of Software Test (AST 2021), held in conjunction with the 43rd International Conference on Software Engineering (ICSE 2021), as well as new original submissions on the theme of "Automatic Software Testing from the Trenches." This issue initially received a total of 13 submissions. Both the extended papers from AST 2021 as well as the new original submissions underwent a rigorous review process, and ultimately, 8 submissions were accepted for inclusion in this special issue.

In the article "Towards practical application of mutation testing in industry — Traditional versus extreme mutation testing," the authors present the results of a case study conducted in a company from the semiconductor industry. A large Java system software project with more than 11,000 unit tests was mutated to analyze traditional and extreme mutation from different perspectives. Among the main findings, the authors report that customer focus, personal priorities, team structure, and the established development processes influence the developers' decision on which mutants should be killed.

The article "An automated search-based test model generation approach for structural testing of model transformations" addresses the challenge of generating valid meta-model instances for model transformation testing in the context of model-driven engineering. The authors propose applying a search-based strategy to generate test models that conform to multiplicity and semantic constraints and further maximize structural coverage of the transformation. The article also describes a testing environment and an empirical evaluation that achieves significance coverage and performance in both a benchmark and an industrial-scale transformation.

The article "Machine learning-based test oracles for performance testing of cyber-physical systems: An industrial case study on elevators dispatching algorithms" presents a test oracle for regression testing called DARIO. This oracle uses machine-learning algorithms to predict test results from previous test data, thus avoiding the expensive running time of traditional regression test oracles. Experiments in an industrial case study show that DARIO reaches competitive accuracy and performance, for both functional and non-functional performance bugs.

In "Predictive maintenance of infrastructure code using "fluid" datasets: An exploratory study on Ansible defect proneness," the authors conducted an empirical study on defect prediction in infrastructure as code scripts. The authors conducted a large empirical evaluation using 50 structural metrics and reported that the naive Bayes classifier achieved the best performance, outperforming the current state of the art.

The article "The effect of hoisting on variants of Hierarchical Delta Debugging" extends the hierarchical delta debugging (HDD) test reduction techniques by enabling the use of transformations other than pruning. The approach replaces subtrees with compatible subtrees further down the hierarchy and extends three baseline HDD variants with hoisting. The experimentation results found that hoisting can help to further reduce the size of test cases by as much as 80% compared with the baseline variants.

"Advanced Security Testing using a Cyber-Attack Forecasting Model: A Case Study of Financial Institutions" focuses on a proactive approach that predicts cyber-attacks to provide new testing scenarios supported by the attack forecasting. The model is based on deep neural networks trained on some of the biggest cyber-attacks on banking institutions over the past 3 years. The performance of the forecasting model was evaluated in a real-life banking environment, and the results show that it provided a high forecasting accuracy.

In the article entitled "AmPyfier: Test Amplification in Python," the authors introduce a test amplification tool to the dynamically typed language Python. AmPyfier uses type profiling to infer the types of methods, variables, and parameters used in the existing test cases. The proposed tool was evaluated on 11 open-source projects, and the results showed that AmPyfier could strengthen around 70% of the test classes.

The article "Model-based generation of test scripts across product variants: An experience report from the railway industry" reports an experience on the collaboration between academia and industry. The article presents a model-based approach to automatically generate test scripts in software product lines (SPL) that is evaluated on the Alstom SA Aventra SPL. The evaluation shows the applicability of the approach in industrial settings and that mitigates the development effort, error proneness and consistency drawbacks of the opportunistic reuse of test artifacts.

We would like to take this opportunity to thank all contributors to this special issue, including all the authors who submitted their work to this call, the reviewers, and the staff of the journal's editorial office. Lastly, we would like to thank the *Journal of Software: Evolution and Process* for publishing this special issue.

Breno Miranda[1]
Javier Tuya[2]
Alejandra Garrido[3]

[1]*Informatics Center, Federal University of Pernambuco, Recife, Brazil*
[2]*Computer Science Department, University of Oviedo, Oviedo, Spain*
[3]*LIFIA, Fac. de Informática, Univ. Nac. de La Plata and CONICET, La Plata, Argentina*

Correspondence
Breno Miranda, Informatics Center, Federal University of Pernambuco, Recife, Brazil.
Email: bafm@cin.ufpe.br