SOFTWARE TESTING, VERIFICATION AND RELIABILITY
Softw. Test. Verif. Reliab. 2019;e1726
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/stvr.1726

Editorial

Special issue on testing extra-functional properties

Co-located with the 10th IEEE International Conference on Software Testing, Verification and Validation (ICST 2017) in Tokyo, we started with and organized the first International Workshop on Testing Extra-Functional Properties and Quality Characteristics of Software Systems (ITEQS)[†].

The importance of having a dedicated forum discussing various aspects of testing EFPs becomes more apparent considering the following points. With the ever-increasing role of computer systems in our daily life, we rely more and more on the services that are provided by a software. As a consequence, the expectations and demands regarding the quality of these services are also dramatically growing. In this context, the success and correctness of a software product may not only be dependent on the logical correctness of its functions but also on their other quality attributes such as performance, security, safety, availability and robustness. Such system characteristics, which are referred to and captured as extra-functional properties (EFPs), or non-functional properties, have determinant importance particularly in resource constrained systems. For instance, in the real-time embedded domain, there can be limitations on available memory, CPU and processing capacity, power consumption and so on, that need to be considered along with timing and security requirements of an application. These systems, therefore, need to be tested with a special attention to EFPs.

Testing a system with respect to its EFPs, however, poses specific challenges, and traditional functional testing methods and approaches may not simply be applicable. Examples of such challenges are fault localization, the need to have appropriate techniques for different types of EFPs, the role and impact of the environment in testing EFPs, observability and testability issues, coverage and test-stop criteria, modelling EFPs and generating meaningful test cases, test oracles for security and privacy that involve hyperproperties, etc.

Considering the peculiarities and challenges of testing EFPs, the main purpose of ITEQS has been to provide a well-focused forum with the goal of bringing together researchers and practitioners to share ideas, identify challenges, propose solutions and techniques, and in general, expand the state-of-the-art and practice in testing EFPs and quality characteristics of software systems.

Since 2017 and until today, ITEQS has been held each year co-located with the ICST conference, attracting different articles and audience discussions, having keynote speeches from well-known researchers in the field, and also panel discussions on specific themes related to the overall topic of the workshop.

This special issue on testing EFPs in the *Software Testing, Verification and Reliability Journal* was established with the ITEQS 2018 workshop, inviting selected best papers from both ITEQS 2017 and 2018 to submit extensions of their work and also being open to any other external high-quality research articles on the topic.

The first paper, “An Exploration of Effective Fuzzing for Side-channel Cache Leakage” by Tiyash Basu, Chundong Wang and Sudipta Chattopadhyay focuses on the problem of validating software systems against both cache timing-based and access-based attacks. They present a coverage metric and a simulated annealing-based test generation approach that explores the cache behaviour. The approach has been evaluated against two state-of-the-art fuzz testing tools in both a

[†] http://www.mrtc.mdh.se/ITEQS/2017/

controlled simulated setting and with real hardware. The results indicate that the proposed approach is more effective in revealing cache side-channel leakage than the other tools.

The second paper, “Reducing the Maintenance Effort for Parameterization of Representative Load Tests Using Annotations” by Henning Schultz, André van Hoorn and Alexander Wert proposes an annotation model to minimize the manual maintenance effort to make generated load tests executable. Stored manual parameterizations for load testing can then evolve automatically as the workload changes, thus significantly reducing the updating effort. The results indicate that the maintenance effort can be reduced to the initial creation of the annotations given that the API does not change.

The third paper, “Leveraging Metamorphic Testing to Automatically Detect Inconsistencies in Code Generator Families” by Mohamed Boussaa, Olivier Barais, Gerson Sunyé and Benoit Baudry presents an approach to test code generators with respect to performance and resource usage of the code generated for different platforms. The authors describe the metamorphic relation as a comparison between the variations of performance and resource usage of test suites running on different versions of the generated code. They apply statistical techniques to define this metamorphic relation and describe a container-based testing environment for deploying, executing and monitoring the resource usage of generated code in multiple target software platforms. The results indicate that the proposed approach is able to automatically detect real issues in code generator families.

Hereby, we would also like to thank all the authors who considered the special issue on testing EFPs in the *Software Testing, Verification and Reliability Journal* for publishing their research articles, and also all the reviewers whose review comments and recommendations helped us to ensure the quality of the special issue and also helped the authors to improve their papers.

Mehrdad Saadatmand
RISE Research Institutes of Sweden, RISE SICS Västerås, Västerås, Sweden

Birgitta Lindström
University of Skövde, School of Informatics, Skövde, Sweden

Bernhard K. Aichernig
Graz University of Technology, Institute for Software Technology, Graz Austria

DOI: 10.1002/stv