The Journal of Systems & Software 203 (2023) 111751

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Editorial

Introduction to the special issue on test automation: Trends, benefits, and costs

Antonia Bertolino, Guglielmo De Angelis, Maurizio Leotta *, Filippo Ricca

ISTI-CNR, Pisa, Italy
IASI-CNR, Rome, Italy
DIBRIS, University of Genova, Genova, Italy
DIBRIS, University of Genova, Genova, Italy

ARTICLE INFO

Article history:
Available online 18 May 2023

Keywords:
Software testing
Automation
Trends
Benefits
Costs
Special issue

It is with a great pleasure that we, as the Guest Editors, can finally present the Virtual Special Issue on "Test Automation: Trends, Benefits, and Costs" for the *Journal of Systems and Software*.

Today software is pervasive in any aspects of our society, and automation across the whole Software Testing process has manifested its powerful contribution to the validation of the systems that software controls. Indeed, research in automating both the synthesis of test cases and their execution has made progress towards meeting the expected adequacy requirements while reducing the high effort and costs that testing activities usually demand.

Notwithstanding the huge research effort that has been invested on testing techniques and tools, though, we find that software developers still lack adequate knowledge and frameworks to quantify the resources that test automation actually demands to the existing software production eco-systems (e.g., technological stacks, but also software companies or human organizations).

In this sense, we launched this special issue in order to explore current research in trends, benefits and costs of test automation that can effectively contribute to the quality assessment of modern software systems. In addition, being aware of the importance about the readiness levels each novel solution has to compare with, we looked for contributions highlighting the opportunities offered by the proposed techniques, while also discussing how to mitigate issues that potentially impact on their broader adoption. Accordingly, we distributed a detailed call for papers, inviting the research community to propose works in test automation regarding approaches, techniques, tools and experience reports about adopting test automation and improving its effectiveness, as well as empirical studies and case studies discussing the costs and benefits of the proposed solutions.

In addition, we recruited a Committee of Distinguished Reviewers, formed by scientists expert in software testing and its automation. The role of this Committee was to ensure a high quality and consistent review process focused on the main objectives of the call. In Table 1 we report the members of the Distinguished Reviewers Committee with their affiliation (when they accepted our invitation). Of course, along the review process we also invited several other external referees to help review the submissions received. To all the committee members and the external reviewers goes our sincere gratitude for the devoted effort that certainly greatly contributed to improve the quality of this Special Issue.

Concerning the visibility and the interest of the research community to the Special Issue, we received 20 scientific papers on the proposed topic, which were submitted between December 2020 and May 2021. Among these, 11 submissions were finally accepted for publication within the Special Issue. The healthy number of manuscripts we received even bypassed our expectations and remarks the interest of the research community in the proposed topic; moreover, most of the submissions were relevant and sound, resulting in a good acceptance rate (i.e., 55%) at the end of a rigorous review process.

* Corresponding editor.
E-mail addresses: antonia.bertolino@isti.cnr.it (A. Bertolino), guglielmo.deangelis@iasi.cnr.it (G.D. Angelis), maurizio.leotta@unige.it (M. Leotta), filippo.ricca@unige.it (F. Ricca).

https://doi.org/10.1016/j.jss.2023.111751

Table 1
Distinguished Reviewers Committee.

Member	Affiliation(s)
Jonathan Bell	Northeastern University (USA)
Lionel Briand	University of Ottawa (Canada) University of Luxembourg (Luxembourg)
Gabriella Carrozza	Accenture Italia (Italy)
Marcelo d'Amorim	Federal University of Pernambuco (Brasil)
Serge Demeyer	University of Antwerp (Belgium)
Sigrid Eldh	Ericsson AB (Sweden)
Emelie Engström	Lund University (Sweden)
Michael Felderer	University of Innsbruck (Austria) Blekinge Institute of Technology (Sweden)
Robert Feldt	Chalmers University of Gothenburg (Sweden) Blekinge Institute of Technology (Sweden)
Moonzoo Kim	Korea Advanced Institute of Science and Technology (South Korea)
Mika Mäntylä	University of Oulu (Finland)
Leonardo Mariani	University of Milano-Bicocca (Italy)
Dusica Marijan	Simula Research Laboratory (Norway)
Mike Papadakis	University of Luxembourg (Luxembourg)
Patrizio Pelliccione	University of L'Aquila (Italy) Chalmers University of Gothenburg (Sweden)
John Penix	Google, Inc (USA)
Silvia R.Vergilio	Federal University of Paraná (Brasil)
Rudolf Ramler	Software Competence Center Hagenberg GmbH (Austria)
Federica Sarro	University College London (UK)
Daniel Sundmark	Mälardalen University (Sweden)
Hong Zhu	Oxford Brookes University (UK)

In the following we provide a brief summary of the papers included in the Virtual Special Issue, with a cordial invitation to go and read the full original texts. We provide a full reference of the papers in the bibliography of this guest introduction.

In the work "Automatically generating test cases for safety-critical software via symbolic execution" (Kurian et al., 2023), Kurian and co-authors address the automated generation of test cases for safety critical systems based on symbolic execution. They observe that some linguistic features commonly banned from critical programs because they are believed to downgrade safety are also the main efficiency-blockers for test generation based on symbolic execution. Hence, they show how by leveraging the same programming constraints adopted for the sake of safety, symbolic execution –if specifically tailored on the those restrictions– can become a viable approach in practice for a significant class of industrial safety-critical systems. In particular, they present the TECS test generator tool for SCADE programs and report their real-world experience in using TECS for testing the on-board signaling unit of high-speed railway software. In this case study, their test suites achieved a high model coverage with good efficiency and, after the manual augmentation of assertion-style test oracles, could also identify some previous unknown faults.

In the article "Diversity-driven unit test generation" (Kessel and Atkinson, 2022), Kessel and Atkinson explore whether providing multiple implementations of the functionality of interest is viable and effective to improve results of automated unit test generation. The idea is to exploit the redundancy available in mainstream software repositories. More specifically, they present and evaluate a hybrid approach that combines a mainstream unit test generation tool with a software search engine to generate or amplify test sets. They show that by exploiting the diversity available in software repositories, such a hybrid approach outperforms the standalone use of test generation tools.

Within the research work "Discovering boundary values of feature-based machine learning classifiers through exploratory datamorphic testing" (Zhu and Bayley, 2022), Zhu and Bayley address some of the features that hinder the testing of AI applications. Building on some previous results, the authors propose and assess three exploratory testing strategies for machine-learning applications in the framework of the datamorphism testing methodology. These techniques aim at exploring the data space of a classification or clustering application to discover the boundaries between the defined classes. In addition to including an extensive theoretical study of the addressed problem, the work has also practical relevance, as it helps testers understand precisely the behavior and function of the software under test. An empirical study of the techniques implemented in the Morphy tool is conducted on a set of controlled experiments and case studies, which evaluated the capability and cost of the proposed test strategies.

In the article "Test automation maturity improves product quality – Quantitative study of open source projects using continuous integration" (Wang et al., 2022), Wang and co-authors investigate the relationship between test automation maturity and continuous integration (CI) success. They survey main contributors and mine some repositories (GitHub repository, CI repository, and issue track system) of 37 java-based open source projects. From the analysis they distill three notable aspects: (a) a potential benefit of improving test automation maturity level using standard best practices in the literature is improving product quality and accelerating the release cycle in the CI context, (b) higher levels of test automation maturity are highly significantly associated with higher product quality and, (c) the effect of test automation maturity is more significant than product size, product complexity, product popularity, product age, and team size on product quality.

Within the research work "A co-evolutionary genetic algorithms approach to detect video game bugs" (Albaghajati and Ahmed, 2022), Albaghajati and Ahmed present a solution for the automated testing of video games software, which can notoriously suffer from different challenging types of bugs due to its inherent complexity, concurrency and non-determinism. The proposed approach relies on the collaborative work of two agents based on genetic algorithms (i.e., co-evolutionary): one agent generates buggy states, and the other performs a reachability analysis of such states using a colored Petri nets representation of the software workflow. The work includes several experiments through which the authors show the potential of the proposed automated approach in effectively finding bugs in different video games systems.

In the article "SuMo: A mutation testing approach and tool for the Ethereum blockchain" (Barboni et al., 2022), Barboni and co-authors propose a novel mutation testing strategy, and a corresponding infrastructure, for assessing the test suites developed to evaluate smart contracts written for the Ethereum blockchain. Mutation testing is *per se* a rather expensive activity that however has been demonstrated to be one of the most effective ways to improve the quality of a test suite. In order to reduce the cost of such an activity, a set of design choices has been made so to reduce, for instance, the generation of possible stillborn mutants (i.e., invalid mutants that generate compilation errors). At the same time, the framework makes it possible the selection of a set of operators so to reduce the number of generated mutants. The authors evaluated the proposed strategy on a wide set of real smart contracts, demonstrating the usefulness of the approach.

In the work "Cost-effective load testing of WebRTC applications" (Gortázar et al., 2022), Gortázar and co-authors investigate novel strategies for non-functional assessment of the real-time video communication standard WebRTC,[1] and present an open source framework that can simplify the load testing WebRTC applications in the cloud. Specifically, for each of the proposed strategies the work studies their performance and costs expressed in terms of number of users emulated, as well as CPU and memory used. In addition, the observed outcomes are also compared with several traditional browser-based strategies.

The contribution titled "Selenium-Jupiter: A JUnit 5 extension for Selenium WebDriver" (García et al., 2022) by García and co-authors proposes a novel open-source framework for end-to-end testing of web applications. Specifically, Selenium-Jupiter is a Java-based toolkit that provides seamless integration between the JUnit 5[2] programming model and the features of Selenium WebDriver,[3] by expanding the latter and reducing the code that has to be written by developers. The work includes an empirical study highlighting the potentials of the proposed framework on realistic industrial settings. However, the empirical validation also reveals few drawbacks of the current version of the framework: the authors critically address these aspects by discussing their motivations and giving precise suggestions on how to investigate them more in detail.

Within the research work "Automated test-based learning and verification of performance models for microservices systems" (Camilli et al., 2022), Camilli and co-authors mitigate performance and scalability issues in applications based on microservices. Specifically, the work leverages load testing sessions that are driven by realistic workload conditions. Starting from individual performance estimations for each microservice, the proposed approach feeds a Bayesian inference model in order to incrementally learn the performance of the whole application under several operational profiles and alternative deployment scenarios sampled from production. The performance model learned during such in vitro load testing sessions is then verified in order to assess if the microservices application under test meets its non-functional requirements. The work includes an empirical study evaluating benefits and costs of the proposed approach against a benchmark application.

The article titled "Scalability testing automation using multivariate characterization and detection of software performance antipatterns" (Avritzer et al., 2022) by Avritzer and co-authors focuses on benefits and costs on the inclusion of a machine learning-based approach for the detection of Software Performance Antipatterns (SPAs) in traditional Continuous Integration (CI) and Continuous Delivery/Deployment (CDD) pipelines. The work identifies parts of system showing the largest impact on scalability by leveraging load testing experiments and multivariate statistical analysis on their outcomes. While the study accepts that both the performance analysis and the SPAs characterization are performed offline, the detection of SPAs is considered an online procedure and the authors provide recommendations for the efficient integration of the latter into the CI/CDD pipelines.

In the article "Automated Web application testing based by pre-recorded test cases" (Sunman et al., 2022), Sunman and co-authors discuss some of the drawbacks of fully automated techniques for testing Web Applications. In this context they propose a pragmatic and semi-automated approach based on exploratory testing practice. Specifically their solution takes as input a set of interaction sequences captured during exploratory testing activities; by means of such configurations the proposed framework automatically generates further test cases by guiding the exploration of the target Web Application. Despite a small reduction of automation, the study highlights that the experience of human testers can be easily exploited in order to improve the effectiveness of the whole testing process.

Before concluding, we would like to thank first the Editors-in-chief of the *Journal of Systems and Software*, Paris Avgeriou and David C. Shepherd, for accepting our proposal, and then Raffaela Mirandola, the assigned "Special Issue Editor", for being always responsive and comprehensive all along the long editorial process. Once again we thank all the many reviewers for their qualified service. And, finally, of course we are grateful to all the authors (of both accepted and rejected papers) who trusted this initiative and consigned their work in our hands.

In our role of Guest Editors we are very happy with the obtained result. In fact, this long journey of collecting scientific papers, coordinating their review process and summarizing them here has led us to discover many interesting research directions and to ascertain, once again, how the automation of Software Testing, in all its facets, is important within the field of software engineering.

We sincerely hope you enjoy the papers in this special issue!

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Albaghajati, A., Ahmed, M., 2022. A co-evolutionary genetic algorithms approach to detect video game bugs. J. Syst. Softw. 188, 111261. http://dx.doi.org/10.1016/j.jss.2022.111261.

Avritzer, A., Britto, R., Trubiani, C., Camilli, M., Janes, A., Russo, B., van Hoorn, A., Heinrich, R., Rapp, M., Henß, J., Chalawadi, R.K., 2022. Scalability testing automation using multivariate characterization and detection of software performance antipatterns. J. Syst. Softw. 193, 111446. http://dx.doi.org/10.1016/j.jss.2022.111446.

Barboni, M., Morichetta, A., Polini, A., 2022. SuMo: A mutation testing approach and tool for the Ethereum blockchain. J. Syst. Softw. 193, 111445. http://dx.doi.org/10.1016/j.jss.2022.111445.

Camilli, M., Janes, A., Russo, B., 2022. Automated test-based learning and verification of performance models for microservices systems. J. Syst. Softw. 187, 111225. http://dx.doi.org/10.1016/j.jss.2022.111225.

García, B., Kloos, C.D., Alario-Hoyos, C., Organero, M.M., 2022. Selenium-Jupiter: A JUnit 5 extension for Selenium WebDriver. J. Syst. Softw. 189, 111298. http://dx.doi.org/10.1016/j.jss.2022.111298.

Gortázar, F., Gallego, M., Maes-Bermejo, M., Chicano-Capelo, I., Santos, C., 2022. Cost-effective load testing of WebRTC applications. J. Syst. Softw. 193, 111439. http://dx.doi.org/10.1016/j.jss.2022.111439.

Kessel, M., Atkinson, C., 2022. Diversity-driven unit test generation. J. Syst. Softw. 193 (C), http://dx.doi.org/10.1016/j.jss.2022.111442.

Kurian, E., Briola, D., Braione, P., Denaro, G., 2023. Automatically generating test cases for safety-critical software via symbolic execution. J. Syst. Softw. 199 (C), (mar) http://dx.doi.org/10.1016/j.jss.2023.111629.

Sunman, N., Soydan, Y., Sözer, H., 2022. Automated web application testing driven by pre-recorded test cases. J. Syst. Softw. 193, 111441. http://dx.doi.org/10.1016/j.jss.2022.111441.

Wang, Y., Mäntylä, M.V., Liu, Z., Markkula, J., 2022. Test automation maturity improves product quality - quantitative study of open source projects using continuous integration. J. Syst. Softw. 188, 111259. http://dx.doi.org/10.1016/j.jss.2022.111259, URL https://www.sciencedirect.com/science/article/pii/S0164121222000280.

Zhu, H., Bayley, I., 2022. Discovering boundary values of feature-based machine learning classifiers through exploratory datamorphic testing. J. Syst. Softw. 187, 111231. http://dx.doi.org/10.1016/j.jss.2022.111231.

[1] See at: https://webrtc.org/.

[2] See at: https://junit.org/junit5/.

[3] See at: https://www.selenium.dev/.