

## EDITORIAL

# Introduction to the special issue: “Software Quality for Modern Systems”

Guglielmo De Angelis<sup>1</sup> | Hyunsook Do<sup>2</sup> | Bao N. Nguyen<sup>3</sup>

<sup>1</sup>IASI-CNR, Rome, Italy

<sup>2</sup>Department of Computer Science and Engineering,  
University of North Texas, Denton, Texas, USA

<sup>3</sup>F5 Inc., California, USA

## Correspondence

Corresponding author Guglielmo De Angelis,  
Email: [guglielmo.deangelis@iasi.cnr.it](mailto:guglielmo.deangelis@iasi.cnr.it)

## Abstract

The volume at hand presents the Special Issue on “Software Quality for Modern Systems”. The Special Issue has been originally conceived within the context of the 3rd International Conference on Automation of Software Test (AST 2022). The authors of the best papers accepted at AST 2022 have been invited to submit an extended version of their previous work. In addition, the editors have decided to keep the submissions open for any other contributions aligned with the objectives of the Special Issue. The main objective of the Guest Editors is to promote discussions leveraging software testing automation for improving various software quality attributes.

## KEY WORDS

software testing, automation, software quality, special issue, AST 2022

It is with a great pleasure that we, as the Guest Editors, can finally present the Special Issue on “Software Quality for Modern Systems” for the Journal of Software: Evolution and Process.

Software pervasiveness in Modern Systems strongly affects both industry and digital society. As a notable recent example, the proliferation of Artificial Intelligence (AI) technologies is continuously leading to emerging needs and expectations from both software producers and consumers. At the same time, infrastructures, software components, and applications aim to hide their increasing complexity in order to appear more human-centric. However, the potential risk from design errors, poor integration, and time-consuming engineering phases can result in unreliable solutions that can barely meet their intended objectives. In this context, Software Engineering processes keep demanding for the investigation of novel and further refined approaches to Software Quality Assurance (SQA).

Software testing automation is a discipline that has produced noteworthy research in the last decades. The search for solutions to automatically test any concept of software is critical, and it encompasses several areas. These include generating test cases, test oracles, and test doubles (e.g., dummies, stubs, mocks, fakes); defining test selection and prioritisation criteria; engineering infrastructures governing and optimising the execution of testing sessions locally or remotely in the cloud.

In this sense, we launched this special issue in order to explore current research methods, empirical evaluations, or industrial case studies involving software test automation for modern software systems such as: AI solutions or applications, Mobile Applications, Adaptive Systems, or Distributed and Cloud Platforms. Accordingly, we crafted a dedicated call for papers, inviting both the research and industrial communities to submit research papers in test automation focusing on improving various software quality attributes.

The Special Issue has been originally conceived within the context of the 3rd International Conference on Automation of Software Test (AST 2022). The authors of the best papers accepted at AST 2022 have been invited to submit an extended version of their previous work. In addition, the editors have decided to keep the submissions open for any other contributions aligned with the objectives of the Special Issue.

The Special Issue attracted the interest of researchers from all over the world: mostly from Europe, but also from Asia, Australia, Middle-East, North Africa and North America. The Special Issue received 7 scientific papers during the submission period between August and December 2022. Among these, 4 submissions were finally accepted for publication, and an article is

an extended version of a work presented at AST 2022. In the following we report a brief resume of the accepted papers, but we also invite the reader to consult the whole contributions. In this sense, please refer the bibliography section of this editorial message.

In the paper “*Understanding the Quality and Evolution of Android App Build Systems*”<sup>1</sup>, Pei Liu et al. describes the methods and results from a mining software repositories study on the usage of build systems in Open Source Android repositories. The authors studied a large dataset of 6,960 projects selected from over 70,000 AndroZooOpen projects. The results from this study provides several interesting empirical insights to both researchers and practitioners including the most popular build system used for Android projects and why; how the dependencies in those projects are often evolved and managed. Those insights are very important to understand how modern software is built and maintained today.

In paper “*Exploring the Context of Use for Voice User Interfaces: Toward Context-dependent UX Quality Testing*”<sup>2</sup>, Klein et al. perform a comprehensive study on how the Voice User Interface (VUI) systems are used. The results provide many useful insights on how modern applications user interfaces should be designed beyond the traditional command line and graphical interfaces. They can be used as a great starting point for future studies on the Human-Computer Interaction model for modern software.

In the work titled “*An empirical study to compare three web test automation approaches: NLP-based, programmable, and capture&replay*”<sup>3</sup>, M. Leotta et al. propose an NLP-based test automation approach in the web context and conducted a series of case studies that compared the proposed approach with two traditional approaches. The results of the study indicate that the NLP-based test automation approach is more competitive for small to medium sized test suites and it reduces the total cumulative cost, which was measured in terms of test cases development and test cases evolution.

Finally, Koitz-Hristov et al. present “*On the Suitability of Checked Coverage and Genetic Parameter Tuning in Test Suite Reduction*”<sup>4</sup>, a Java test suite reduction framework that allows to compute minimised test suites considering different code coverage metrics ranging from method coverage, line coverage, to the more recently introduced checked coverage. An empirical evaluation was performed by comparing the three implemented test suite minimisation algorithms (i.e., Greedy HGS, Delayed Greedy, and a genetic algorithm), considering the three different code coverage metrics and assessing the time required for the test suite reduction framework. The work is an extension of a paper previously accepted and presented at AST 2022.

As final remarks, the Guest Editors would like to thank the whole board of Editors-in-chief of the Journal of Software: Evolution and Process. Among the others, we thank Massimiliano Di Penta for his work and for being always responsive and comprehensive all along the long editorial process. We also feel to acknowledge all the many reviewers for their qualified service for the community, and all the authors that with their work supported this initiative.

## References

1. Liu P, Li L, Liu K, McIntosh S, Grundy J. Understanding the quality and evolution of Android app build systems. *Journal of Software: Evolution and Process*.;n/a(n/a):e2602. doi: <https://doi.org/10.1002/smr.2602>
2. Klein AM, Deutschländer J, Kölln K, Rauschenberger M, Escalona MJ. Exploring the context of use for voice user interfaces: Toward context-dependent user experience quality testing. *Journal of Software: Evolution and Process*.;n/a(n/a):e2618. doi: <https://doi.org/10.1002/smr.2618>
3. Leotta M, Ricca F, Marchetto A, Olinas D. An empirical study to compare three web test automation approaches: NLP-based, programmable, and capture&replay. *Journal of Software: Evolution and Process*.;n/a(n/a):e2606. doi: <https://doi.org/10.1002/smr.2606>
4. Koitz-Hristov R, Sterner T, Stracke L, Wotawa F. On the Suitability of Checked Coverage and Genetic Parameter Tuning in Test Suite Reduction. *Journal of Software: Evolution and Process*. –not-appeared-yet.