Software Quality Journal (2020) 28:369–370
https://doi.org/10.1007/s11219-020-09518-x

# In this issue

Rachel Harrison[1]

Published online: 26 May 2020

I hope that all our readers are healthy and safe at this time of global uncertainty.

In this issue, we have four regular research papers and three special sections.

The first two regular research papers are linked by the common theme of software testing, whereas the final two regular papers report secondary research, the first being a survey of APIs and the second a systematic literature review concerned with refactoring.

Testing is a crucial stage of the software development lifecycle. In “Increasing Validity Through Replication: An Illustrative TDD Case” , Adrian Santos, Sira Vegas, Fernando Uyaguari, Oscar Dieste, Burak Turhan, and Natalia Juristo draw attention to the importance of replication when performing software engineering experiments and also the agile software development process of test-driven development. The authors conclude that replication may improve the reliability of the previous findings and help to overcome the previous experiments’ weaknesses.

Continuing with the theme of agile software development, the paper “Enhanced Regression Testing Technique for Agile Software Development and Continuous Integration Strategies” by Sadia Ali, Yaser Hafeez, Shariq Hussain, and Shunkun Yang proposes a test case prioritization and selection approach that was validated using an empirical study. The results show that the approach can enhance the regression testing process for agile development.

In “What Are the Characteristics of Popular APIs? A Large-Scale Study on Java, Android, and 165 Libraries”, Caroline Lima and Andre Hora examine the characteristics of popular APIs. The authors report that popular APIs are different in that they tend to be larger, have more comments, and be more unstable than unpopular APIs. They also suggest some factors that developers can use to improve APIs.

Systematic literature reviews can provide interesting insights of how a field has evolved over time. The paper “Automatic Software Refactoring: A Systematic Literature Review” by Abdulrahman Ahmed Bobakr Baqais and Mohammad Alshayeb provides an overview of papers that propose or implement an automated refactoring process. They found that most papers discuss code refactoring, with only a few recent papers focusing on model refactoring. Search-based refactoring is also now gaining popularity.

✉ Rachel Harrison
rachel.harrison@brookes.ac.uk

[1] School of Engineering, Computing and Mathematics, Oxford Brookes University, Oxford OX33 1HX, UK

The first special section is concerned with advanced software applications. I would like to thank the guest editors, Lefteris Angelis, and Tomas Bures, for all their hard work on bringing this section to fruition. They have provided a helpful introduction to this special section.

The second special section concerns formal methods. The guest editors, Yliès Falcone and Leonardo Mariani, have worked hard to bring this section to press. They have also provided an introduction to this special section to guide your reading.

The final special section is on testing and repair. I am indebted to the guest editors, Ian Bayley, Yan Cai, and Patricia Machado, for soliciting and editing this section. Again they have kindly provided an introduction to their special section.

As always, I am grateful for your suggestions or comments on this issue; please email me at rachel.harrison@brookes.ac.uk.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.