



In this issue

Rachel Harrison¹

Published online: 31 August 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

In this issue, we have a special section and eight regular research papers. The special section is on trustworthy systems and software. I am very grateful to the guest editors, Sudipto Ghosh and Zhenyu Chen, for all their hard work on this special section. The guest editors have provided a helpful introduction to this special section to guide your reading.

The first two regular research papers are linked by the common theme of code smells, and these are followed by a systematic literature review on risk factors found in software development. We then have two papers on the performance of developers followed by two papers on testing and one on the quality of modelling languages.

In “A large-scale empirical study of code smells in JavaScript projects”, David Johannes, Foutse Khomh, and Giuliano Antoniol describe a large-scale study of JavaScript code smells to better understand how they impact the fault-proneness of applications. The results show that code smells do affect the quality of JavaScript applications negatively. The authors suggest that developers should track and remove smells early in the software life cycle.

Systematic mapping studies help to highlight the strengths and weaknesses of a research field. The paper “Software Design Smell Detection: a systematic mapping study” by Khalid Alkharabsheh, Yania Crespo, Esperanza Manso, and José A. Taboada analyzes 18 years of research into design smell detection. From the 395 papers analyzed, the authors report that there is a lack of human expertise and benchmark validation processes and also show that design smell detection positively influences quality attributes. They suggest that it would be helpful to have a reference repository of design smells labeled by experts.

Systematic literature reviews are similar to mapping studies but usually follow rigorous protocols to provide a very detailed analysis of the literature. In “Risk factors in software development projects: a systematic literature review”, Júlio Menezes Jr., Cristine Gusmão, and Hermano Moura identify and map risk factors found in software development project environments. The authors conducted a systematic literature review and categorized 148 different risk factors. The results show that risk factors related to software requirements are frequently cited, together with a lack of technical skill.

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

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

The performance of developers is of particular concern to managers and can be a problematic issue. In “To var or not to var: how do C# developers use and misuse implicit and explicit typing?” Pierre A. Akiki looks at the difference between implicit and explicit typing in C# and provides an overview of developers’ opinions and of the guidelines that are available online. The paper reports on an analysis of the source code of 10 different open-source software projects with more than 16,500 thousand lines of code. The paper also presents a tool called Code Analysis and Refactoring Engine for C# (Care#). Future work includes an extension of Care# to support more types of analysis and refactoring.

The Personal Software Process can help developers to improve their performance. The paper “Assisting software engineering students in analyzing their performance in software development” by Mushtaq Raza, João Pascoal Faria, and Rafael Salazar describes a tool for automated experiment performance analysis and improvement recommendations. The authors performed a controlled experiment involving 61 software engineering students, half of which used the new tool in a Personal Software Process (PSP) performance analysis assignment, while the other half used a traditional PSP support tool for performing the same assignment. The results showed significant benefits in terms of the students’ satisfaction and the time required to do the analysis. In future work, the authors will investigate the application of the new tool for analyzing the performance of teams adhering to agile practices.

Testing is of paramount importance to the industry. In “Classifying generated white-box tests: an exploratory study”, Dávid Honfi and Zoltán Micskei describe exploratory studies to investigate the performance of developers during white-box test analysis. The studies were carried out in a laboratory setting with 106 graduate students. The results showed that participants do tend to incorrectly classify tests. The authors suggest using a conceptual framework to describe the classification task.

Continuing the theme of testing, the paper “An efficient regression testing approach for PHP Web applications using test selection and reusable constraints” by Ravi Eda and Hyunsook Do discusses an approach to test selection for PHP Web applications where a subset of existing tests that cover the modified code paths can be detected. The authors use a tool to identify tests that can be reused with a new software version. Results show that this approach is effective in reducing the cost of regression testing.

The final paper in this issue is concerned with the quality of modelling languages. In “A method to evaluate quality of modelling languages based on the Zachman reference taxonomy”, Fábio D. Giraldo, Sergio España, William J. Giraldo, Óscar Pastor, and John Krogstie propose using principles from an information systems architecture reference (the Zachman framework) as a taxonomy for modelling languages. The paper derives formal, methodological, and technological requirements for a modelling language quality evaluation method to tackle some of the open model-driven engineering quality challenges. In the future, the authors will improve the tool by adding visualization options and populate the tool with more examples of taxonomic analysis.

I hope that you will find this issue interesting and informative. As usual, if you have any suggestions or comments 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.