Software Qual J (2018) 26:657–660
https://doi.org/10.1007/s11219-018-9406-x

Special issue on "software quality in software-intensive systems"

Emilia Mendes[1,2] · **Dietmar Winkler**[3]

Published online: 19 April 2018

This Special Issue looks at articles that investigate any software quality aspects within the context of software-intensive systems. The focus upon quality is due to its pivotal role when developing and managing any software solution, which also applies to software-intensive systems. Such systems are described as "any system where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole" [IEEE Std 42010:2011]. Examples of software-intensive systems are embedded systems for avionics and automotive applications, large-scale heterogeneous systems, or business applications with special focus on Web services.

The special issue has papers arranged within three main topics, as follows:

1. Model-based development, components, and services
2. Software quality management: measurement, peopleware, and innovation
3. Software process and product improvement

We received a total of 16 manuscripts, of which five were accepted for publication. Each manuscript was reviewed by at least two reviewers.

The five papers that were accepted and are therefore part of this Special Issue are briefly introduced next, and they are grouped according to one of the three main abovementioned topics they belong to:

Topic 1: Model-based development, components, and services

Paper: Architecture optimization: speed or accuracy? Both!; by Federico Ciccozzi, Juraj Feljan, Jan Carlson, and Ivica Crnkovic

✉ Emilia Mendes
emilia_mendes@yahoo.com

[1] Blekinge Institute of Technology, Karlskrona, Sweden

[2] The University of Oulu, Oulu, Finland

[3] Vienna University of Technology, Vienna, Austria

Motivated by the increase in the complexity of embedded systems, and the need for early analysis of sound architectural solutions with respect to quality attributes, the authors propose a new method for architecture optimization that combines both a model-based and an execution-based approach. Its main principle is to first employ model-based optimization for a fast identification of a good architecture candidate, which is then used as input for an execution-based optimization, which is a slower however more accurate type of optimization. Their results showed that the proposed solution identified, on average, a better architectural solution, when compared to the solution that would be proposed using solely a pure execution-based or pure model-based optimizations.

Paper: Considerations about quality in model-driven engineering: current state, challenges, and example; by Faber D. Giraldo, Sergio España, Oscar Pastor, and William J. Giraldo

The motivation for this work relates to the lack of consensus in the model-driven engineering (MDE) field, on what constitutes quality within that context. The authors detail the current state of the art relating to quality in MDE, where the various definitions of quality are organized into 16 categories; further, they also highlight quality issues in MDE's real applications. Using 73 studies as a basis, they argue there exists a mismatch between quality evaluation of modeling languages as investigated by academia and the actual MDE industrial practice. Another contribution of this work is to put forward eight challenges for MDE projects' quality assessment that current initiatives do not yet address.

Topic 2: Software quality management: measurement, peopleware, and innovation

Paper: Coherence of comments and method implementations: a dataset and an empirical investigation; by Anna Corazza, Valerio Maggio, and Giuseppe Scanniello

The motivation of this study relates to whether the lead comments of a method, written by developers while coding, genuinely describe methods and actual implementations' intent, i.e., are lead comments of methods coherent with their source code?. To investigate the topic, the authors have gathered data on 3636 methods in 3 open source software applications implemented in Java and did a manual assessment of coherence between comments and source code. Such data is stored in a dataset that is now publicly available too. They have also carried out an exploratory investigation to check whether support vector machines could be used to help discriminate between coherent and non-coherent methods, using lexical similarity as basis.

Topic 3: Software process and product improvement

Paper: Challenges of software process and product quality improvement: catalyzing defect root-cause investigation by process enactment data analysis; by Mehmet Söylemez and Ayca Tarhan

The motivation for this research relates to improving both the software development process and product quality by understanding the root causes of defects, and to be able to do this even if an organization is emergent or resides at lower maturity levels. The authors present and assess a method that employs root-cause analysis to understand the root cause of software defects. Defects were gathered from a software-intensive project of a CMMI ML3-certified institute. The proposed method combines process enactment data collection and analysis with a defect root-cause analysis technique called Orthogonal Defect Classification (ODC). Results showed that the method could be applied efficiently, and led to improvements in development performance and product quality.

Paper: Integrating software quality models into risk-based testing; by Harald Foidl and Michael Felderer

The motivation for the last paper in this special issue relates to combining a quality model-based quality assessment with risk-based testing. The authors look at this issue by first detailing two different and general ways to integrate quality assessments based on quality models with risk-based testing. This is followed up by a specific integration, with tool support, that uses the open quality model QuaMoCo as a basis to illustrate how quality assessment and risk-based testing can work together. This integration is later assessed in a case study where five open source products are used. Results showed that the risk-based strategy outperformed the sole use of a code-based testing strategy, as per the number of classes that should be tested in order to find all defects. Further, they also found a relationship between the risk coefficient and the associated number of defects of a class.

We do hope that you will be pleased when reading the papers in this Special Issue, and we would like to add that this would not have been possible without the great help and support of a number of people, as follows: the authors who submitted excellent papers to the Special Issue; all the reviewers who did a brilliant job in selecting very high-quality contributions, which led to the selection of the five papers detailed herein; and last but not the least, the SQJ editor-in-chief Professor Rachel Harrison, for agreeing to include this Special Issue, and for her guidance throughout the whole process.

Emilia Mendes is Full Professor in Computer Science at the Blekinge Institute of Technology (Sweden), and also a Tekes-funded Finnish Distinguished Professor at the University of Oulu (Finland). Her areas of research interest are mainly within the context of empirical software engineering, value-based software engineering, and the use of machine learning techniques to contexts such as healthcare, and sustainability. She has published widely and over 200 refereed papers, plus two books as solo author – both in the area of cost estimation. She is on the editorial board of several journals, which include TSE and the SQJ.

Dietmar Winkler is a senior researcher at the Information and Software Engineering Group at TU Wien, Austria. He is currently working in the Christian Doppler Laboratory for "Security and Quality Improvement in the Production System Lifecycle" (CDL-SQI) at the faculty of Informatics at TU Wien. His research interests include software and systems engineering process and product improvement in multidisciplinary engineering environments, quality management and quality assurance, as well as empirical evaluations in industrial settings.