Software Qual J (2016) 24:203–204
DOI 10.1007/s11219-016-9313-y

In this issue

Rachel Harrison[1]

Published online: 24 March 2016

In this issue we have nine regular research papers. The first four papers are all concerned with aspects of quality models and quality attributes, whereas the next four all look at various aspects of software testing. The final paper in his issue is concerned with software process models, particularly hybrid agile models.

In "The COCA quality model for user documentation", Bartosz Alchimowicz and Jerzy Nawrocki propose a new quality model for user documentation together with a set of acceptance criteria. The model has just four quality characteristics (Completeness, Operability, Correctness, and Appearance). The model has been validated using nine user manuals taken from the educational software domain.

High-quality user documentation must be readable, and this is also an important attribute of high-quality software. The paper "Beauty and the Beast—On the Readability of Object-Oriented Example Programs", by Jürgen Börstler, Michael E Caspersen, and Marie Nordström discusses the quality attributes of simplicity and understandability from a cognitive perspective. The authors show how to measure these attributes and validate the measures using programs taken from 12 popular introductory Java programming textbooks. It is suggested that readability is a sub-factor of understandability, and a measure, the Software Readability Ease Score, is proposed and compared with other readability measures with good results. The measure could be used to give students immediate feedback on the readability of their programs.

In "Software quality improvement: a model based on managing factors impacting software quality", Ivan Janicijevic, Maja Krsmanovic, Nedeljko Zivkovic, and Sasa Lazarevic propose a systematic framework for modelling the processes of a quality management system stochastically and show how to select the optimum set of factors for software quality improvement. The methodology allows managers to identify those areas in which efforts to improve quality will be particularly effective.

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

[1] Department of Computing and Communication Technologies, Oxford Brookes University, Oxford OX33 1HX, UK

Continuing the theme of quality models and attributes, the paper “A survey on quality attributes in service-based systems” by David Ameller, Matthias Galster, Paris Avgeriou, and Xavier Franch, examines the role of quality attributes such as dependability in the design of service-based systems. The authors carried out a survey with practitioners in industry and found that most quality attributes are addressed by ad hoc means. They also found that dependability and performance are considered to be particularly important attributes for service-based systems. It is suggested that focussing on these attributes in future may reap the greatest rewards.

The final paper on this theme looks at the important nonfunctional requirement of usability. In “A model-driven approach for usability engineering of interactive systems”, Lassaad Ben Ammar, Abdelwaheb Trabelsi, and Adel Mahfoudhi describe a method for evaluating usability throughout a model-driven engineering development life cycle. The authors propose the use of usability-driven model transformations and early usability evaluation. The approach is validated via a case study.

Turning our attention to testing, the paper “Adoption and Use of Cloud-Based Testing in Practice”, by Leah Riungu-Kalliosaari, Ossi Taipale, Kari Smolander, and Ita Richardson discusses the adoption, use, and effect of cloud-based testing in various organizational contexts. The authors carried out interviews with professionals in 20 organizations and found that that cloud-based testing can provide a solution to testing needs within an organizational context. They propose a strategy that can help organisations during adoption of cloud-based testing.

In “Testing variability-intensive systems using automated analysis: an application to Android”, José A. Galindo, Hamilton Turner, David Benavides, and Jules White discuss testing for software product lines. They describe how to use a constraint satisfaction solver to prune, prioritize, and package product line tests. They validate their approach using an example which shows the benefits of maximizing the mobile market share while meeting a budgetary constraint.

Continuing on the theme of testing, the paper “Analysis of distance functions for similarity-based test suite reduction in the context of model-based testing”, by Ana Emilia Victor Barbosa Coutinho, Emanuela Gadelha Cartaxo, and Patricia Duarte de Lima Machado investigates test suite reduction in the context of model-based testing. This is important when test cases have been generated automatically. The authors discuss the effectiveness of distance functions for reduction strategies based on test case similarity. Results show that the choice of distance function has little influence on the size of the reduced test suite but can significantly affect fault coverage.

The final paper in this issue is concerned with software process models. In “Prioritizing agile benefits and limitations in relation to practice usage”, Adam Solinski and Kai Petersen investigate hybrid process models for software development, with a view to seeing how agile processes are used in practice. Agile processes were found to have a number of benefits, including learning, developer satisfaction, and social skill development. Disadvantages of agile process adoption were found to include skill-specific demands and scalability. Some agile practices did not lead to the desired benefits. In particular, pair-programming, test-driven development, and continuous integration with testing were sometimes abandoned.

As usual, if you have any thoughts on this issue, please send them to me via: rachel.harrison@brookes.ac.uk.