Software Qual J (2015) 23:537–538
DOI 10.1007/s11219-015-9293-3

In this issue

Rachel Harrison[1]

Published online: 5 October 2015

In this issue, we have six regular research papers which are all concerned with aspects of software development and software quality.

Global software development has become an integral part of the software industry over recent years, but such development is not without problems. In "Handover of managerial responsibilities in global software development: a case study of source code evolution and quality", Ronald Jabangwe, Jürgen Börstler, and Kai Petersen discuss the quality of two products that were developed using global software development. They discovered that software project management responsibilities can be passed between two geographically dispersed sites without a negative impact on quality. They also found that development work can be carried out using such dispersed sites without adversely affecting quality. These results are important because it had previously been thought that distributing software development around the world may have a detrimental effect on quality.

Continuing with the global software development theme, the paper "An experience-based framework for evaluating alignment of software quality goals" by Panagiota Chatzipetrou, Lefteris Angelis, Sebastian Barney, and Claes Wohlin considers the agreement between stakeholders when prioritizing goals for a project. The authors propose a framework of techniques to study prioritization alignment and then apply this framework to an empirical study of global software development.

Another relatively new concept in the field of information and communication technology is that of cloud computing. In "A framework and tool to manage Cloud Computing service quality", Francisco Jose Dominguez-Mayo, Julian Alberto Garcia-Garcia, Maria Jose Escalona, Manuel Mejias, Matias Urbieta, and Gustavo Rossi describe a capability maturity model for offshore outsourcing services. The model can be used to assess capability models and standards in the cloud as well as software processes and products.

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

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

In "Software quality construction in 11 companies: an empirical study using the grounded theory", Frank Philip Seth, Erja Mustonen-Ollila, Ossi Taipale, and Kari Smolander discuss the role of human factors in software quality management. The authors used a grounded theory approach to analyze software development activities and methods. Forms of cooperation between teams of developers were also compared. The authors conclude that software quality arises as the result of a number of human factors both inside and outside software development companies.

The paper "Spoiled patterns: how to extend the GoF" by Cedric Bouhours, Herve Leblanc, and Christian Percebois introduces a new concept called *spoiled patterns*. In a spoiled pattern, some parts of the original pattern may be present but these parts might not be properly connected. The authors describe how they collected spoiled patterns, and compare spoiled patterns with structural variations, bad smells, and anti-patterns. They suggest that spoiled patterns may be useful in computer science education, showing how patterns may emerge as approved solutions over time.

In the final paper of this issue, "Empirical analysis of factors affecting confirmation bias levels of software engineers", Gul Calikli and Ayse Bener examine the tendency of human beings to seek confirmation rather than negation of their beliefs. This natural bias can lead to inefficiencies throughout the software development process. The authors identify a number of factors that can increase such bias. They conclude that individuals who have been trained in logical reasoning and mathematical proof techniques are more likely to refute a statement than to immediately accept it. They also found that software development or software testing experience does not seem to have an effect on confirmation bias. The authors suggest that organizations should find ways to improve logical reasoning and hypothesis-testing skills among their software engineers if they want to reduce confirmation bias.

As usual, please send any thoughts you may have on this issue to me (rachel.harrison@brookes.ac.uk).