

In this issue

Rachel Harrison

Published online: 23 November 2010
© Springer Science+Business Media, LLC 2010

This issue includes a special section based on extended papers that were originally presented at the 2009 International Conference on Program Comprehension (ICPC 2009). The guest editors of this special section, Rainer Koschke, Andrian Marcus and Gerald C. Gannod, provide an introduction to these selected papers. I would like to thank the guest editors for all their hard work in managing the selection and reviews of these papers.

The five regular papers in this issue deal with a range of issues including process modelling, e-learning repositories, software testing, effort estimation and software quality modelling.

In “Towards objective software process information: experiences from a case study”, Jos Trienekens, Jana Samalikova, Rob Kusters, Ton Weijters and Paul Siemons discuss the differences between actual and documented software process modelling. The authors show how process models which reflect reality more accurately can be constructed automatically by software process mining. The paper presents details of how software process mining was applied to a number of case studies in a large company in The Netherlands. This paper shows how software process improvement can be discovered as a result of software process mining.

Nowadays the web is seen as an invaluable source of reference material for almost all aspects of human endeavour. However, the quality of the material which is available cannot be assured, and this is potentially extremely problematic. The paper “Metrics-based evaluation of learning object reusability” by Javier Sanz-Rodriguez, Juan Manuel Dodero and Salvador Sanchez-Alonso discusses the selection of reusable educational materials from the web. The authors propose a way to measure the reusability of learning objects. Two e-learning repositories are used to validate the metric. This work may result in a measure of overall quality for web repositories.

Testing is a crucially important part of the software development process. In “Contributions of tester experience and a checklist guideline to the identification of categories and choices for software testing”, Pak-Lok Poon, T.H. Tse, Sau-Fun Tang, and Fei-Ching Kuo discuss methodologies for the generation of test suites. The authors investigated the types

R. Harrison (✉)

School of Technology, Oxford Brookes University, Wheatley, Oxford OX33 1HX, UK
e-mail: rachel.harrison@brookes.ac.uk

and numbers of mistakes made by both inexperienced and experienced software testers whilst using an ad-hoc approach to test suite generation and the improvements that could be made by discussing the mistakes with testers and by giving them a checklist to use during test case selection. Three specifications taken from the real-world were used for the investigation. The use of the proposed checklist could help to increase the effectiveness of test case selection.

The paper “Comparison of weighted grey relational analysis for software effort estimation” by Chao-Jung Hsu and Chin-Yu Huang describes some of the work that has been done on project effort estimation. The authors propose the use of a similarity-based method, grey relational analysis, for effort estimation. In particular, the paper presents six weighted methods that can be used with conventional grey relational analysis. The performance of these methods was tested using four public data sets. The authors conclude that grey relational analysis can be used for software effort estimation. The method may become a useful addition to the project manager’s toolkit.

In “Graphical versus textual software measurement modelling: an empirical study”, Beatriz Mora, Félix García, Francisco Ruiz, and Mario Piattini introduce a graphical domain specific language which can be used to define software quality models. This Software Measurement Modelling Language was validated empirically via experiments using students as subjects in a classroom environment. The authors conclude that it is useful to have a suitable notation with which to define software quality models.

Please let me know if you have any suggestions or comments on this issue: I can be emailed at rachel.harrison@brookes.ac.uk.