Guest Editorial for Special Section from Empirical Software Engineering & Measurement (ESEM) 2011

In this section we present five specially selected and extended articles from the 5th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement which took place in Banff, Canada in September 2011. This conference attracted 103 full papers from which we selected 33 for presentation and publication, an acceptance rate of 32%. From these, authors of five of the most highly rated papers were invited to extend their papers for inclusion in this special issue which is intended to act as a showcase for our research to the wider software engineering community.

ESEM2011 is the fifth in a series of Symposia that aim to provide a forum where researchers and practitioners can report and discuss recent research results in the areas of empirical software engineering and software measurement. They encourage the exchange of ideas that help communicate the strengths and weaknesses of software engineering technologies and methods from an empirical viewpoint and are widely regarded as the premier international event for empirical software engineering researchers, investigators and interested practitioners. The five papers included address one or both themes of (i) presenting research results from high quality primary studies and (ii) exploring strengths and weaknesses of competing and emerging research methods.

Paper summaries:

Andrew Austin, Casper Holmgreen, and Laurie Williams. “A Comparison of the Efficiency and Effectiveness of Vulnerability Discovery Techniques”.

Security of software systems, particularly for health systems where there are strong confidentiality and privacy issues, is increasingly being recognized as important. Consequently a number of vulnerability detection techniques have been advocated. Unfortunately there is remarkably little rigorous empirical evidence concerning their efficacy. Thus this primary study is timely and welcome. Here the authors describe three case studies based on three E-health record systems to assess four different vulnerability discovery techniques (exploratory manual penetration testing, systematic manual penetration testing, automated penetration testing, and automated static analysis). They find that no one technique was able to discover all threats but that a combination of systematic manual penetration testing and automated static analysis could provide a good level of coverage.

This paper is an excellent example of empirical research that focuses on topical issues that are of interest and significance to the software practitioners.


This is another example of a high quality primary study that investigates the recently growing and not well understood end-user software development technology. The researchers investigated 32,000 mashups from the Yahoo! Pipes repository and its associated community of end-user developers. One of the striking findings was the existence of many near duplicate artifacts suggesting that there is a need for improved end-user programmer support for development, maintenance, search, and program understanding.

Again it is gratifying to find empirical software engineering researchers broadening their reach and addressing non-traditional software development that nevertheless have large user bases and have considerable impact economically and societally.

Sebastian Nanz, Faraz Torshizi, Michela Pedroni and Bertrand Meyer. “Design of an Empirical Study for Comparing the Usability of Concurrent Programming Languages”.

This paper is interesting in that it not only constitutes a primary study that investigates relative usability of different programming languages that support concurrency but also presents a framework to assist the conduct of future primary studies. As the authors observe, there is something of a proliferation of programming languages to support concurrency yet we have little hard evidence regarding their respective merits and demerits. In their study the authors compare multithreaded java and SCOOP but because there are many other languages and the need for further investigation also address the issue of providing support for future empirical researchers. An interesting challenge is the balance between pragmatism and the ‘perfect’ empirical study. Clearly we wish to avoid a situation where pursuit of the latter results in a dearth of empirical evidence. The paper presents a simple, easy to accomplish approach that could be accommodated within the constraints of a typical undergraduate programming course.


This is a further paper that is both a useful primary study in its own right but also has explicit regard for follow up and replication studies. The study is a mix-method study, that is it combines both
qualitative and quantitative analysis to extend our empirical understanding of how team composition impacts team performance in software development. It is also the culmination of five years of studies. Many of the phenomena we study, including software team composition and performance, are complex and so these long term research programmes are apposite. It is also impressive to see how the authors explicitly integrate the various stages of their research programme into a more powerful whole.


The final paper in this Special Section differs from the other four in that it does not embody a primary study but is instead a tertiary study, that empirically analyses the adoption, value, and use of systematic reviews (or secondary studies) in empirical software engineering. Similarly to the da Silva et al. paper a mixed methods approach is adopted, this time based on a systematic review of secondary studies, questionnaires and interviews. Zhang and Babar located almost 150 systematic reviews since 2004 so clearly this is a popular and significant research instrument and therefore obtaining a better appreciation of the issues is valuable. Interestingly a dominant concern is the practical one of getting the balance between resources available and methodological ‘purity’.

We would like to thank the authors, the reviewers and the Editor-in Chief of Information & Software Technology for their hard work and support that has made this Special Section possible. We hope you enjoy it.

Martin Shepperd*
Brunel University, United Kingdom
Tel.: +44 0123456789.
E-mail address: martin.shepperd@brunel.ac.uk

Forrest Shull
Fraunhofer USA, Centre for Experimental Software Engineering, United States

Available online 7 January 2013