



A Focus on the Human Side of Software Engineering

MARIAN PETRE

m.petre@open.ac.uk

Faculty of Mathematics and Computing, The Open University, Milton Keynes, MK7 6AA, UK

DAVID BUDGEN

d.budgen@cs.keele.ac.uk

School of Computing and Mathematics, Keele University, Keele, Staffordshire, ST5 5BG, UK

JEAN SCHOLTZ

jean.scholtz@nist.gov

Visualization and Usability Group, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA

Keywords: Psychology of programming, empirical studies, EASE, PPIG.

The Evaluation and Assessment in Software Engineering (EASE) and Psychology of Programming Interest Group (PPIG) are both “grass roots” communities, one motivated by software engineers, the other by cognitive psychologists. Both examine theoretical and methodological issues in software development—but from different perspectives and traditions. Questions about software development, ranging from fundamental issues of problem solving and representation, to highly applied issues of development processes and software maintenance, are complex and inherently interdisciplinary, making it essential that they be addressed from a variety of scientific perspectives.

The EASE and PPIG constituencies have long sought ways to interact in order to broaden the discourse and embrace multiple perspectives. EASE and PPIG co-located their annual conferences in April 2003 at Keele University, supported by a grant from the UK’s Engineering and Physical Sciences Research Council (EPSRC). The aim was to take advantage of the co-location to promote joint activities targeted at improving cross-fertilization of ideas between the two groups, and through them between software engineering and psychology of programming more generally. Cross-fertilization provides leverage to bring industrial relevance to psychology of programming pursuits, and to bring psychology of programming insights into empirical software engineering, thus contributing both to well-grounded theory about software engineering processes and to well-informed software development practice.

The EASE Community (<http://ease.cs.keele.ac.uk>)

The first EASE Conference was held at Keele in 1997, and was initiated as a collaborative effort by a number of software engineers from Keele University (Prof.

David Budgen, Prof. Barbara Kitchenham, Prof. Pearl Brereton and Mr. Stephen Linkman). Since then it has continued to run annually, with nominal support from the British Computer Society (BCS) and Institute of Electrical Engineers (IEE), and with an ever-widening international community of participants. The EASE tradition is multi-disciplinary and addresses both methodological issues and experience, drawing papers from industry and academia.

The PPIG Community (<http://www.ppig.org/>)

The PPIG arose from *ad hoc* meetings of researchers brought together by common interests. It was established as a “group” by Prof. T. R. G. Green in 1987 in order to provide a focus for people from diverse communities who were exploring psychological aspects of programming. “Programming” is interpreted in the broadest sense to include any aspect of software development, at any level. PPIG quickly attracted international interest, and the group currently numbers approximately 300 worldwide, including (but not limited to) cognitive scientists, psychologists, computer scientists, software engineers, software developers and HCI researchers in both academia and industry.

The five papers in this special issue are a selection of expanded topics from the co-located conferences, chosen for their consonance with the editorial mission of this journal:

Ahonen, Junttila, and Sakkinen, in “Impacts of the Organizational Model on Testing: Three Industrial Cases”, derived process models from documentation and elicitative interviews in three organizations in order to identify issues related to testing and suggestions for improvement. They observed that testing does tend to be a weak link in the software engineering process, but that the main problems relate to the design of test cases and to organizational challenges, rather than to the testing infrastructure.

Jørgensen, in “Regression Models of Software Development Effort Estimation Accuracy and Bias”, addresses the capacity of regression models to explain the accuracy and bias variation of an organization’s estimates of software development effort. Drawing on data from 49 software development tasks, he found that, despite careful modeling, “the explanatory and predictive power of both models was poor”. Jørgensen suggests that regression-based models are limited in the extent to which they can encompass all important estimation accuracy and bias factors, but that such models may nevertheless be useful as support for human judgment.

Moløkken and Jørgensen, in “Group Processes in Software Effort Estimation”, describe an experiment in which experts with different backgrounds combined their estimates of software development effort in unstructured group discussions, producing group estimates which were “less optimistic and more realistic” than either individual estimates or the averages of individual estimates. They suggest that discussion, a relatively cheap and easily transported method, reveals activities and complexities not considered or previously under-estimated.

Müller, in “Are Reviews an Alternative to Pair Programming?”, reports an experiment comparing pair programming and individual programming augmented with a code review phase, in terms of code quality and development cost. Although the study is limited by its use of student subjects and relatively small programming tasks, its outcomes warrant further study: the two treatments produced roughly the same quality with roughly the same cost.

Robinson and Sharp, in “An Ethnographic Study of XP Practice”, present an ethnographic study of a mature XP team, revealing that the “reality of practice” is more sophisticated, holistic, and socially embedded than the 12 prescriptive practices of XP portray. They give insight into the social activity that underpins XP practice, and show that an oral tradition of knowledge sharing can be used successfully to support software development.

The five papers cross the spectrum of software development activities, from estimation, through development processes and productivity, to testing. They also show a range of approaches to the empirical study of software development, from controlled experimentation and formal modeling through ethnographic study—several showing an interplay between qualitative and quantitative analyses. Each provides some insight, not just into software development, but also into the richness afforded by a multiple-method, multiple-perspective approach to its investigation.



Marian Petre is Reader in Computing at the Open University and an EPSRC Advanced Research Fellow investigating “Expert behavior and reasoning in the design of complex systems”. Her research aims to contribute to a theory of engineering expertise, based on empirical study of actual, situated, expert practice, and hence to contribute to better tools and education for designers.



David Budgen is a Professor of Software Engineering at Keele University, UK. He received BSc and PhD degrees in theoretical physics from the University of Durham in 1969 and 1973. After a period working on naval command and control systems, where he developed an interest in software design issues, he was

appointed to a post at the University of Stirling, moving to a chair of software engineering at Keele in 1991. The second edition of his book *Software Design* was published in 2003. His main research interests include software design, component based and service based technologies, software development environments, and empirical software engineering.



Dr. Jean Scholtz is on the editorial board of *Empirical Studies of Software Engineering*. She is a researcher at the National Institute of Standards and Technology where her interests are in developing metrics and methodologies for evaluating interactive systems. Her current work focuses on evaluation of information systems for intelligence analysts and on human–robot interaction systems, including search and rescue and on-road driving. She holds a PhD in Computer Science from the University of Nebraska.