

## Qualitative research in software engineering

Tore Dybå • Rafael Prikładnicki • Kari Rönkkö •  
Carolyn Seaman • Jonathan Sillito

Published online: 28 May 2011  
© Springer Science+Business Media, LLC 2011

Qualitative research methods were developed in the social sciences to enable researchers to study social and cultural phenomena and are designed to help researchers understand people and the social and cultural contexts within which they live (Denzin and Lincoln 2011). The goal of understanding a phenomenon from the point of view of the participants and its particular social and institutional context is largely lost when textual data are quantified. Taylor and Bogdan (1984) point out that qualitative research methods were designed mostly by educational researchers and other social scientists to study the complexities of human behavior (e.g., motivation, communication, difficulties in understanding). According to these authors, human behavior is clearly a phenomenon that, due to its complexity, requires qualitative methods to be fully understood, since much of human behavior cannot be adequately described and explained through statistics and other quantitative methods. Examples of qualitative methods are action research, case study research, ethnography, and grounded theory. Qualitative data sources include observation and participant observation (fieldwork), interviews and questionnaires, documents and texts, and the researcher's impressions and reactions.

Many in the software industry recognize that software development also presents a number of unique management and organizational issues that need to be addressed and solved in order for the field to progress. And this situation has led to studies related not only

---

T. Dybå  
SINTEF, Trondheim, Norway

R. Prikładnicki  
PUCRS, Porto Alegre, Brazil

K. Rönkkö  
Blekinge Institute of Technology, Karlskrona, Sweden

C. Seaman (✉)  
University of Maryland Baltimore County, Baltimore, MD, USA  
e-mail: cseaman@umbc.edu

J. Sillito  
University of Calgary, Calgary, Alberta, Canada

to software engineering technical issues, but also to non-technical issues as well as to the intersection between the technical and non-technical aspects. Complex software engineering issues can be difficult to study using a purely quantitative approach. The reasons for this include typically small sample sizes, the expense of controlled experiments with human subjects, and the need for preliminary support before hypothesis testing can begin. Alternatively, qualitative studies can generate well-grounded hypotheses and findings that incorporate the complexity of the phenomenon under study. They also offer richer explanations and new areas for future study. They are also appropriate when variables are not defined or quantified and there is little prior theoretical or empirical work.

Lastly, a principal advantage of using qualitative methods is that they force the researcher to delve into the complexity of the problem rather than abstract it away, and the results can be more informative. Thus, since software engineering involves complex human behavior in an environment and circumstances that are to date not well developed theoretically or empirically and the phenomenon requires rich explanation in order to develop constructs and hypothesis, we need to raise the awareness regarding qualitative research in this field.

For this reason, we solicited submissions for this Special Issue on the topic of “Qualitative Research in Software Engineering”. This special issue was meant to overcome some of the methodological challenges by providing software engineering researchers with good examples of the state of the art in the application of qualitative research methods to software engineering problems. We hoped to include articles that could illustrate good flexible research designs, and how methodological challenges in software engineering research can be addressed.

In this special issue we have gathered four papers that present innovative ideas on how to cope with the growing complexity of effectively researching software engineering phenomena using qualitative approaches. We received papers from around the world, 24 in total. Each paper was reviewed by at least three expert referees. The guest editors would like to thank all the reviewers and authors of submitted papers to this Special Issue.

The four papers of this Special Issue employ a variety of qualitative research methods to address a variety of software engineering phenomena. Two of the papers, those by Prechelt and Oezbek and by Adolph et al., employ grounded theory, although they both tailor and interpret traditional grounded theory techniques for the particular needs of software engineering researchers. Thus, they provide a much-needed resource for researchers in our field who are struggling with adapting grounded theory based on resources written for researchers in education and the social sciences. Sim and Alspaugh’s paper on their “war story” study of requirements engineering is another example of tailoring a traditional qualitative research method (interviews) to the context of studying software engineering phenomena. Another twist in a well-known research approach, the case study, is presented in McLeod et al., who present a detailed explanation and rationale for a “longitudinal” case study methodology. Synopses of these four papers follow.

Prechelt and Oezbek present a narrative about their multiple attempts to study the phenomenon of software process improvement in open source software development. They found this subject very difficult to address, and tried a number of well-reasoned approaches to the problem. In addition to the approach that was finally successful, an adaptation of grounded theory, their article provides a rare exposition of the failings of several other methodological approaches to this problem.

McLeod et al. give a detailed descriptive and prescriptive treatment of a type of case study approach, the longitudinal case study, which is particularly well suited for exploring complex software development phenomena. In particular, the authors found this approach

useful in understanding the multiple factors, including time, that interacted to effect the outcomes of the project being studied.

Sim and Alspaugh provide deep insights and five valuable recommendations related to analyzing data from “war stories” interviews. Standard quantitative analysis techniques, which broke down the stories into smaller “facts”, led the authors to lose the structure of each story. On the other hand, analysis using techniques from the humanities, along with concepts such as methodical vs. amethodical perspectives, enabled new useful insights.

Adolph, et al., present their own experiences of implementing Glaserian grounded theory. Although the method is time consuming and tedious, they find it to be an effective tool for software engineering research. Based on their own case of applying grounded theory in a software engineering context they provide insightful lessons learned and useful guidelines for how to interpret the canons of classical grounded theory literature.

It is our hope that this Special Issue will serve as a resource for software engineering researchers, through its detailed examples of how these authors tailored and applied qualitative research methods to contribute to the body of knowledge in our field. We also sincerely desire for this issue, and for the growing body of literature describing qualitative research in software engineering, to motivate the research community to employ qualitative methods, alone or in mixed methods designs, to more fully study the complexities of software engineering. We believe that progress in our field requires full consideration of human behavior within software development environments, and that understanding the complexities of human behavior requires that researchers go beyond the limitations of quantification and statistical analysis. Qualitative research methods have been found, in many other fields, to be critical to grasping and understanding the phenomena in which humans play a role. We are pleased to present this Special Issue as a contribution towards the continuing need for qualitative research in software engineering.

## References

- Denzin NK, Lincoln YS (2011) *The SAGE handbook of qualitative research*, 4th edn. Sage Publications, Thousand Oaks, California
- Taylor SJ, Bogdan R (1984) *Introduction to qualitative research methods*. John Wiley & Sons, New York



**Tore Dybå** is chief scientist and research manager at SINTEF and adjunct full professor at the University of Oslo. He received his doctoral degree in computer and information science from the Norwegian University of Science and Technology. His research interests include empirical software engineering, software process improvement, and agile software development. His research has been extensively published in major

software engineering conferences and journals including IEEE Transactions on Software Engineering, IEEE Software, Information and Software Technology, Empirical Software Engineering and Software Process: Improvement and Practice. He is the principal author of the book *Process Improvement in Practice: A Handbook for IT Companies* and co-author of *Agile Software Development: Current Research and Future Directions*, both published by Springer. Dr. Dybå was keynote speaker at SBQS 2010 and EASE 2011. He is member of the editorial board of *Empirical Software Engineering* and associate editor-in-chief of *IEEE Software*.



**Rafael Prikladnicki** is a Professor in the Computer Science School at the Pontificia Universidade do Rio Grande do Sul (PUCRS), Brazil. Professor Prikladnicki's areas of expertise are distributed software development and agile methodologies. His 2007 book "Distributed Software Development - developing software with distributed teams" was the first Portuguese book on this topic. He also leads one of the main research groups in this area in Brazil (MuNDDoS research group), and is especially interested in how global software engineering (GSE) interplays with agile methodologies, how GSE impacts organizational decisions and the role of Brazil in the global IT industry.



**Kari Rönkkö** is an Associate Professor of software engineering with emphasis on human and cooperative aspects, at the School of Computing at Blekinge Institute of Technology (BTH) in Sweden. His research interests include flexible research approaches, human communication, ethnomethodology, interaction design, and user experience. Rönkkö has conducted eight years of method development and practice focused action research with industry players. He has led the research group Use-Oriented Design and Development, and is one of the initiators of the Blekinge Engineering Software Qualities Research Center at BTH. Dr. Rönkkö has fifteen years of industrial experience outside academia. He is today part of the research group Game Systems

and Interaction Research Laboratory at BTH, and the Cybernorms research group at faculty of Social Sciences, Lund University.



**Carolyn Seaman** is an Associate Professor of Information Systems at the University of Maryland Baltimore County (UMBC). Her research generally falls under the umbrella of empirical studies of software engineering, with particular emphases on maintenance, organizational structure, communication, measurement, COTS-based development, and qualitative research methods. Dr. Seaman is also a Research Fellow at the Fraunhofer Center for Experimental Software Engineering, Maryland, where she participates in research on experience management in software engineering organizations and software metrics. She holds a PhD in Computer Science from the University of Maryland, College Park, a MS in Information and Computer Science from Georgia Tech, and a BA in Computer Science and Mathematics from the College of Wooster (Ohio). She has worked in the software industry as a software engineer and consultant, and has conducted most of her research in industrial and governmental settings (e.g. IBM Canada Ltd., NASA, Xerox).



**Jonathan Sillito** is an Assistant Professor in the Department of Computer Science at the University of Calgary in Canada. His research is concerned with the process of creating and managing source code changes. He has also created Saturate, a web application for qualitative analysis. Jonathan has a PhD from the University of British Columbia and an MSc and BSc from the University of Alberta.