Using Ethnography and Discourse Analysis to Study Software Engineering Practices

Helen Sharp
Centre for HCI Design
City University
Northampton Square
London EC1V 0HB, UK
+44 20 7477 8481
h.c.sharp@soi.city.ac.uk

Mark Woodman
School of Computing Science
Middlesex University
Trent Park, Bramley Road
London N14 4YZ,
m.woodman@mdx.ac.uk

Hugh Robinson
Computing Department
The Open University
Walton Hall
Milton Keynes MK7 6AA, UK
h.m.robinson@open.ac.uk

ABSTRACT
In 1994, we started a project called SoFEA to investigate the non-technical factors influencing the adoption and evolution of Software Quality Management Systems (SQMS). At that time, various quality initiatives existed including standards, codes of best practice and SQMSs. These focused primarily on technical fixes, yet from anecdotal evidence it seemed clear that many of the problems associated with software quality were non-technical. To investigate these non-technical factors, we employed a combined approach of ethnography and discourse analysis, under the guidance of our social science colleagues. This paper provides some brief background to the project and reports on our experience (as software engineers) of using these techniques.

Keywords
ethnography, discourse analysis, culture, software quality

1 SOFTWARE QUALITY
Software quality and the impact of deficient software has been the subject of some very public debates. Most recently the issues surrounding 'the millenium bug' have raised considerable attention. Prior to this, examples include the London Ambulance system [15], and the Therac-25 accidents [16]. When we started our work back in 1994, software quality initiatives had focused mainly on technical solutions despite the fact that the importance of 'people' and human factors issues in software development had been recognised for some considerable time, e.g. [4]. Evidence from papers and discussions at software quality conferences indicated that these technical initiatives were not providing the increase in quality that was needed. It seemed to us that discussions were focusing on non-technical issues such as organisational culture and acceptance of management initiatives by developers, rather than technical problems.

We therefore embarked on a project to investigate the non-technical factors affecting the adoption and evolution of Software Quality Management Systems [11].

2 OUR APPROACH: COMBINING ETHNOGRAPHY AND DISCOURSE ANALYSIS
Empirical techniques which have been used to study software developers range from broad field studies to individual experiments (e.g. [3, 14, 19]). Since we wished to capture a

holistic view of the context for software quality, and to study the parties and their interactions, we turned our attention to techniques traditionally associated with the social sciences: ethnography and discourse analysis.

The approach we adopted is a hybrid one, with ethnography [7] at its heart. Ethnography is a broad-based approach in which researchers observe their collaborators without prejudice or prior assumptions. Engaging with software developers while they are working on real problems, probably under pressure, will reveal more about their attitudes to software quality than talking to them in abstract terms, or talking to their line managers. Discourse analysis is concerned with the collection and analysis of spoken or written data. It considers what people do with words and how use of these words achieves certain ends.

One of the criticisms of 'classical' ethnography is that it fails to feed back into practice [6]. To address this, others have developed modifications to the basic ethnographic approach (for example [1] and [8]).

In our project, we used ethnography to provide a framework for studying the culture of a setting, and for uncovering the knowledge, ideas, beliefs and values which inform activity. We used discourse analysis to focus on the structures of text to explore people's interactions and interpretations, and uncover the ways in which knowledge, ideas, beliefs and values are transmitted, discussed, developed and perpetuated. Combining ethnography and discourse analysis in one approach means that you do not divorce the texts from their cultural setting, but use the cultural setting to provide enhanced insight and understanding.

3 THE SOFEA PROJECT

The particular focus of the project was on the effectiveness of quality initiatives such as those promoted by quality 'gurus', e.g. [2, 7, 13] and accreditation to ISO9000 [12], and the social factors affecting their adoption, evolution and success. This involved us in visiting five companies of different sizes to investigate stories about their SQMSs.

Initial contact with our collaborators was made through a software quality management conference which resulted in a series of visits to different companies and one one-week long study. In the end, our interlocutors were one person from each of three companies, three people in one company, and all members of a department involved in software development in another company. The two main techniques used were informal interview and participative observation based on ethnographic theory and practice. We used discourse analysis to study the transcripts of the interviews as well as the ethnographic observations. The interviews were semi-structured insofar as we had a list of 'core' topics that we wished interlocutors to cover, but we made the tone as informal and conversational as possible and did not adhere to a pre-determined order.

The data collected consists of audio and video recordings, items from noticeboards, technical documentation, marketing brochures, observations of physical environment and employees' habits, etc.

Results
Results from the project fall into two categories:

1. The factors we found to be influencing the effectiveness of SQMSs.
2. Observations on the use of the chosen techniques from ourselves as software engineers trying to apply them, and from our collaborators as software engineers being 'subjected' to them.

Influences we observed included organisational factors, such as the nature of the company's business, the customer base, market pressures, and individual factors such as a strong community culture in 'Quality', maverick developers, and pressures on individuals. We do not elaborate on these in this paper. More detailed results on this aspect of the project, are reported elsewhere [9, 10, 18].

4 EXPECTATIONS AND EXPERIENCE FROM A SOFTWARE ENGINEER'S PERSPECTIVE

We were new to this style of investigation, and although we were sympathetic to a more human-centred approach to software engineering, we still found aspects of the techniques difficult to come to terms with. Our collaborators too found our approach, and the nature of our results somewhat surprising, although they regarded the findings generally to be 'fairly accurate'. Our activities prompted much discussion within the companies around their SQMS and internal practices, in particular in the company where we stayed for one week. Drawing on our own reactions and those of our collaborators, we briefly describe five aspects which were particularly noticeable.

Seeing the facts for the discourse
We used discourse analysis to uncover some of the tacit ways in which messages are communicated. This can reveal beliefs and values within a culture. However, the speaker is also communicating facts, and as software engineers we are far more used to collecting, analysing, interpreting and using facts. At times, it was difficult to 'see the wood for the trees' and to know how to deal with the facts, and how to deal with the discourse. The following is taken from a discussion we had within our team which expresses how one of us was feeling at the time:

> "… we take meanings of words, we hear what people say and how often they refer to things and the context in which they refer to things and then we also know something about their company and where they individually sit within the company… it's marrying that context with what they are telling us <which is difficult>"

In this respect, it may have been our familiarity with the domain causing us to miss important clues from out interlocutors. To illustrate this point further, when some of our data was taken to a regular discourse discussion group within the University consisting of social scientists, quite different interpretations were ascribed to the discourse, leading to discussions of power and gender. To us these seemed to be inappropriate interpretations, but maybe we were too much a part of the culture we were studying, despite our attempts to remain 'distant'.

Pre-conceptions
We were concerned to allow our collaborators to 'speak for themselves', i.e. we did not ask pre-determined questions aimed at eliciting specific kinds of answer. Our collaborators, particularly, were surprised by this. Following a report to one of them, one comment we received back was:

> "… you kick off by saying that specific questions about software quality were not asked – this makes it harder to put your comments into context."

One of the reasons for not asking software quality questions directly was precisely so that we could ascertain how many of the employees mentioned quality issues without being prompted, and in what context any comments arose. However, in planning for our visits we also found it difficult not to set a quality agenda for our discussions.

Avoiding judgment
Throughout the study, and subsequent analysis, we maintained a non-judgmental position, and treated everything as 'strange'. From our experience we believe that this approach allowed our collaborators to express their own viewpoints more fully, and allowed us to gain a better inside view of the cultures and influences at play. It is important to maintain this stance, and to treat the collaborator as the expert. Maintaining this perspective, from a researcher's point of view, is not easy, and it was sometimes difficult not to comment on some of our observations. For example, one of our interlocutors talked about "*the* software lifecycle" as being something that they had now adopted. This implied a belief that there was only one lifecycle. In a normal conversation with a fellow software engineer, this observation would have been questioned, but we needed to maintain an impartial stance.

This also led to confusion within the team, and concerns about our own language and whether we were using appropriate or inappropriate wording to express our ideas. Knowing how to construct discourse about our observations became very difficult.

What am I doing here?
A related aspect is the lack of hypothesis to test. It is impossible to suspend entirely who you are, your training, your experiences and your expectations. Pragmatically, the best that can be done is to be aware of them and to question how you interpret what you see and hear. Having no hypothesis means that it is easy to feel uncertain of what you are looking for. Of course, you are looking for everything; all comments, events, glances, etc. are significant. During the course of one meeting we videoed, one of us became very uncertain about what to look at, what to take notes about, and what to observe.

The kind of experience with the discourse discussion group described above added to this feeling of uncertainty, and to questions of whether we were applying the technique correctly.

Communicating our findings
Our collaborators had expected us to be judgmental and comparative. They wanted us to be experts, consultants. They regarded us as 'one of them' because we shared a common background, being software engineers with knowledge of quality management issues. This allowed us to enter into informed discussion, but our observations needed to remain

impartial, and our ability to enter into such discussion may have led them into thinking that we would provide different feedback. For example, again after a report to one of the collaborators, they commented:

> "The penultimate paragraph… talks about not making judgements. We feel that a more judgmental approach would have been useful to us… We don't feel that you've drawn many comparisons between the experiences at <the company> and those which you have come across in other parts of the industry… This comparative information, …, would be informative and we think would prove valuable."

Another comment related to the issue of quantitative and qualitative data:

> "… you frequently use 'some' or 'several' which doesn't really give us a feel for the numbers of people who support a given premise. It is quite difficult to decide which comments we should attach importance to."

One of the principles of our approach was that all views should be attended to, and given equal weight, but this was obviously something again which puzzled our collaborators.

5 CONCLUSION

We have been applying social science methods (ethnography and discourse analysis) to uncover how quality procedures are actually applied by practitioners. In this role, we have found the techniques to be powerful and very revealing (a theme explored in more depth in [17]). However, as software engineers applying them in our own discipline, we would suggest that the non-judgmental approach be maintained only until the analysis of data is complete. After this, the specialist expertise of the researchers, e.g. in process improvement or software engineering, should be brought to bear to provide the kind of comparative feedback our collaborators would have appreciated.

We have found that software engineers, both as researchers and as collaborators, have some uncertainties about these methods, and hence are hesitant to embrace them as useful tools. Exposing these uncertainties and discussing ways to overcome them will, we hope, lead to their acceptance by the software community.

ACKNOWLEDGEMENTS

We are indebted to our former colleagues Fiona Hovenden and Hughie McKay for sharing with us their expertise and in helping us to understand our own discipline from this new perspective.

REFERENCES

1. Ball, L. J., & Ormerod, T. C. (1999) 'Applying ethnography in the analysis and support of expertise in engineering design', *Design Studies*, in press.

2. Crosby, P. B. (1979), *Quality is Free*, McGraw-Hill.

3. Curtis, B., Krasner, H. and Iscoe, N. (1988) 'A field study of the software design process for large systems', *Communications of the ACM*, **31**(11), 1268–87.

4. DeMarco, T. and Lister, T. (1987) *Peopleware: productive projects and teams*, Dorset

House Publishing.

5. Deming, W.E. (1982) *Out of the Crisis*, MIT.
6. Hammersley, M. (1992) *What's wrong with ethnography?,* Routledge, London.
7. Hammersley, M. & Atkinson, P. (1983) *Ethnography: principles in practice*, Tavistock, London.
8. Holtzblatt, K. & Beyer, H. (1993) 'Making Customer-Centred design work for teams', *Communications of the ACM*, **36**(10), 93-103.
9. Hovenden, F.M., Sharp, H.C. & Woodman, M. (1998) "Vulcans Versus Humans: The Tension Between Software Quality Management Systems And Software Developers", The Open University, Computing Department Technical Report No. 98/13, 1998.
10. Hovenden, F.M., Walker, S., Sharp, H.C. and Woodman, M. (1996) 'Building quality into scientific software', *The Software Quality Journal,* Chapman & Hall.
11. Hovenden, F.M., Yates, S., Sharp, H.C. & Woodman, M. (1994) 'The Use of Ethnography with Discourse Analysis in the Study of Software Quality Management Systems', in *Software Quality Management II Vol 1: Managing Quality Systems*, M. Ross, C.A. Brebbia, G. Staples and J. Stapleton (eds), Computational Mechanics Publications, pp 557–572.
12. ISO 9000 family of standards (also known as ISO EN BS 9000)
13. Juran, J.M. et al (eds) (1979) *Quality Control Handbook*, 3rd ed McGraw-Hill.
14. Kant, E. (1985) 'Understanding and Automating Algorithm Design', *IEEE Transactions on Software Engineering* **SE-11**(11), November.
15. LAS (1993) Report of the Inquiry into The London Ambulance service, February, available from BDO Consulting.
16. Leveson, N. and Turner, C. (1993) 'An investigation of the Therac-25 accidents', *IEEE Computer*, July, 18-41.
17. Sharp, H, Robinson, H. and Woodman, M. (2000) 'Software Engineering: Community and Culture', *IEEE Software*, **17**(1) 40-47.
18. Sharp, H.C., Woodman, M, Hovenden, F. and Robinson, H. (1999) 'The role of culture in successful software process improvement', in *EUROMICRO '99*, *Procs 25th EUROMICRO Conference,* Milan Italy, 8-10 September 1999, Vol II pp170-6, IEEE Press.
19. Visser, W. (1991) The cognitive psychology viewpoint on design: examples from empirical studies, in *Proceedings of Artificial Intelligence in Design '91*, Gero, J. (ed), pp 505–24, Butterworth-Heinemann, Oxford.

BIOGRAPHIES

Helen Sharp is a senior lecturer in the Centre for HCI Design, a research centre in the School of Informatics at City University. Her research focuses on understanding how software engineers develop software, and hence how they can best be supported in their tasks. In pursuing this goal, she has been involved with a variety of workplace and experimental studies focusing on designers and software engineers.

She is an affiliate of the IEEE Computer Society, and a member of the British Computer Society and ACM. She is also a chartered engineer. Helen received a BSc in Mathematics and an MSc and PhD in Computer Science from University College London.

Hugh Robinson is a senior lecturer in computing at the Open University. He is also a member of the Academic Board of the United States Open University. His research interests center around the nature of computing as an intellectual discipline, using techniques from ethnography and discourse analysis.

Robinson received his BA in Philosophy and American Studies from Keele University, an M.Sc. in computer science from Birkbeck College, University of London and a PhD in computer science from the Council for National Academic Awards, studying at Hatfield Polytechnic.

Mark Woodman is the Professor in Information Technology at Middlesex University, London. His research interests include the design and implementation of complex software systems and the development of appropriate tools for learning, doing, and documenting design. Within this context he researches the social aspects of software development.

Woodman received his BSc in Computer Science from Queens University, Belfast, and his PhD from the Open University. He is a member of the British Computer Society and the ACM.