

# Understanding the Role of Configuration Management Systems in Software Development

Rebecca E. Grinter

Department of Information and Computer Science  
University of California, Irvine  
Irvine, CA 92717-3425  
Tel: +1-714-824-5086  
E-mail: beki@ics.uci.edu

## ABSTRACT

This paper describes a study of how software developers use a technology, software configuration management systems, in their work. The study uses qualitative methods in three case studies to find out how well configuration management systems support the coordination of software development work. Results from this study will help to inform the design of technologies that support group work and provide insights into the complexities of software development.

## Keywords

CSCW, collaborative work, software engineering, empirical studies, development tools, group work, configuration management systems.

## INTRODUCTION

Software development is a complex activity involving many people working together. Researchers in the software engineering community have addressed this complexity in part by developing technologies to support individuals and groups in their work. Despite some successes, software engineering researchers have faced challenges in making these technologies usable and useful to software developers.

Concerns about usability and usefulness give software engineers and HCI researchers a common ground. Recently, these two communities have started to work together on defining these shared interests [10]. My study explores how software developers use one particular technology, configuration management (CM) systems, to coordinate their work. Modern software systems usually consist of hundreds of individual components like: software, libraries, documentation, and test suites. Keeping track of how multiple developers change all of these components during development is very difficult. CM systems provide automated support for maintaining control over the evolution of a software system by: structuring work of developers, providing visibility into

the work of others, and gathering all the system's components together.

My study takes critical look at how developers use CM systems in practice. Specifically the research question is: how, and when, do software engineers use configuration management systems to try to coordinate their software development activities with each other? This study contributes to the establishment of shared concerns by bringing a HCI perspective to understanding the work of software development and the role of technology within that work.

## CONTEXT FOR RESEARCH

Researchers have begun to examine how groupware systems are used in organizations [1, 5, 7]. They have reported on a number of general challenges that users of these tools face in trying to make them work. Among these issues are: the relationship between people's understanding of a technology and its use, the discrepancies between who does the work and who gets the benefit, and clashes between existing organizational structures and the use of groupware.

Software configuration management systems provide another venue to study these issues in detail. However, the developers of software configuration management systems have been relatively isolated from the groupware community. As a consequence, configuration management systems differ from more "traditional" groupware systems. Many traditional groupware systems, like electronic mail, video conferencing, and media spaces, support collaboration by providing mediums for communication. Configuration management systems do not provide any communications support. Instead, they try to support collaboration by providing information about what other developers are doing, structuring development work, and automating various development activities. Because configuration management systems structure software development activities, they are similar to workflow systems. An empirical study of a technology that supports collaboration by providing information to help developers coordinate with each other, rather than another communications channel, raises new questions about the role of technology in supporting group work.

Because people use configuration management systems some of these challenges must be reconciled, at least in the local work context. However, empirical studies of software

© Copyright on this material is held by the author(s)

configuration management systems have revealed that their usage depends on the surrounding organizational and social context [4]. This study has begun and will continue to identify how developers try to use configuration management systems to collaborate with others and the times when the technology fails to support coordination.

## METHODS

The paradigm of qualitative research has been used with legitimacy for some time in the HCI community to conduct exploratory studies (e.g., [2, 6]). Qualitative research provides a suitable approach to studying how software developers use configuration management tools to try to coordinate their software development efforts because it supports exploratory research that provides information about practice. As well as creating rich understandings of configuration management systems work in practice, this qualitative research will generate variables and hypotheses for further quantitative research.

In this study I adopt one particular qualitative data interpretation strategy, grounded theory [3]. Grounded theory suits this study because it meshes perfectly with the theoretical perspective of articulation work that focuses the data collection towards coordination issues. Grounded theory is not entirely inductive and can leverage from existing theoretical bases provided that the theory it relies on has also been developed in the grounded theory tradition. I will use the theory of articulation work, developed in the grounded theory tradition, to focus my study [9].

Articulation work concerns itself not with the primary work of individuals, but the "extra" work they take on in their efforts to coordinate with other people. In the case of software development, developers primarily work on building systems. However, as developers work in a team, they necessarily engage in articulation work to coordinate their own efforts with other people. Unlike other kinds of groupware, CM systems do not facilitate articulation work by providing communications support. Instead CM systems allow developers to coordinate their work without communicating with others. However, little empirical work has been done to date to find out how well technologies can support articulation work [8]. This study will provide valuable information about the role of CM systems in coordinating software development activities.

I have chosen three organizations to conduct this research with the aim of gathering diverse data to strengthen my understanding of configuration management systems usage practices. The three sites I have chosen vary on a number of key dimensions: size of the organization, size of the software project studied, and market that the organization competes in. I use variety of qualitative strategies to collect data including: participant observation, non-participant observation, interviewing and document analysis. By using a combination of techniques in a variety of organizations I have maximized my chances of gathering diverse data.

## CONCLUSIONS AND STATUS

This study explores the role of configuration management systems in coordinating the work of software developers.

The results from the study will contribute to our understanding of how well CM systems help software developers coordinate their work. The results will also provide valuable information about how technologies that do not provide communications facilities support coordination. Finally, this study will present insights into the complexities of software development work.

The data from the three sites has been gathered and is currently being analyzed. Some results are currently available and most or all should be ready by April, 1996.

## ACKNOWLEDGMENTS

I would like to thank the Engineering and Physical Sciences Research Council of the United Kingdom for financial support. I would also like to thank my advisor, Jonathan Grudin.

## REFERENCES

1. Bowers, J. (1994). The Work to Make a Network Work: Studying CSCW in Action. In *Proceedings of CSCW '94*, (22-26 Oct., Chapel Hill, North Carolina), ACM Press, 287-298.
2. Bowers, J., and Pycok, J. (1994). Talking Through Design: Requirements and Resistance in Cooperative Prototyping. In *Proceedings of CHI'94* (24-28 Apr., Boston), ACM Press, 299-305.
3. Glaser, B. G., and A. L. Strauss. (1967), *The Discovery of Grounded Theory: Strategies for Qualitative Research*. New York: Aldine de Gruyter.
4. Grinter, R. (1995). Using a Configuration Management Tool to Coordinate Software Development. In *Proceedings of COOCS*, (13-16 Aug., Milpitas, California), ACM Press, 168-177.
5. Grudin, J. (1989). Why Groupware Applications Fail: Problems in Design and Evaluation. *Office: Technology and People*. 4(3), 245-264.
6. Harper, R., & Sellen, A. (1995). Collaborative Tools and the Practicalities of Professional Work at the International Monetary Fund. In *Proceedings of CHI '95*, (7-11 May, Denver), ACM Press, 122-129.
7. Orlikowski, W. J. (1992). Learning from Notes: Organizational Issues in Groupware Implementation, in *Proceedings of CSCW '92*, (31 Oct. - 4 Nov., Toronto), ACM Press, 362-369.
8. Schmidt, K. and L. Bannon. (1992). Taking CSCW Seriously: Supporting Articulation Work. *Computer Supported Cooperative Work: An International Journal*. 1(1-2), 7-40.
9. Strauss, A. (1985). Work and the Division of Labor. *The Sociological Quarterly*. 26(1), 1-19.
10. Taylor, R. N. and J. Coutaz. (1995). *Software Engineering and Human-Computer Interaction: ICSE '94 Workshop on SE-HCI Joint Research Issues*. Heidelberg: Springer-Verlag.