Qualitative Methods for Studying Distributed Software Development: Issues and Challenges

Sameer Patil
School of Informatics and Computing, Indiana University
901 E 10th Street, Bloomington IN 47408, USA
patil@indiana.edu

1. BACKGROUND AND EXPERIENCE

I was introduced to qualitative methods during the beginning stages of my doctoral studies via a formal course on qualitative research methods as well as an extensive reading list covering various qualitative techniques and interpretive perspectives. Subsequently, I have applied the knowledge and insights by incorporating qualitative methods in my research endeavors. Specifically, I have engaged in semi-structured interviewing, non-participant observations in the field, debrief interviews conducted at the end of experiments and system evaluation studies, and coding of open-ended responses in online questionnaires. In a variety of these projects, I followed a mixed-methods approach by using both qualitative and quantitative approaches in complementary ways (see e.g. [2]). In some cases, both qualitative and quantitative investigations were part of the same study (see e.g., [1, 3]); qualitative analysis not only helped explain some of the quantitative findings but also generated additional insights not available from the numeric data. Moreover, as part of courses I taught on User-driven System Design and Value-Sensitive Design, I introduced students to the theory and practice of some of these techniques.

2. FIELD STUDY OF DISTRIBUTED SOFTWARE DEVELOPMENT

Within Human-Computer Interaction (HCI), my research focus has been on systems for enabling and supporting collaborative work. In particular, I investigated the often-experienced tension between benefits of information disclosure and desires for managing privacy. On the one hand, collaborators gain benefits – collectively as well as individually – by providing colleagues with information about activities, practices, and preferences. On the other hand, disclosure of such information creates potential risks for violations of privacy. Both the benefits as well as the risks are typically higher when collaborators are geographically and/or temporally distributed and the work activities are less tightly-coupled compared with synchronous collaboration. Consequently, in such collaborative projects, the tensions and conflicts between the benefits and the risks are also likely to be more salient. Therefore, distributed projects serve as a particularly interesting case for investigation of these issues.

One of the most commonly studied domains of distributed collaboration is distributed software development. This type of collaborative endeavors received attention from Computer-Supported Cooperative Work (CSCW) as well as software engineering research communities. We, therefore, engaged in an extensive field study of a globally-distributed corporate software development project in order to investigate how collaborators experience and resolve the tension between information disclosure and privacy risks in their daily work practices. The project – “Project X” – involved approximately 125 members spread across five different geographical locations of a large multinational telecommunications corporation: four in the U.S. and one in India. The project was tasked with building a middleware platform that would provide its services in the form of a well-defined API (Application Programming Interface). The goal was to utilize the API as the framework underlying every higher-level application software built by the corporation. As a result, the task of Project X was not only complex and challenging, but it also held organization-wide significance due to the involvement of a variety of stakeholders.

From the beginning, discussions about methodological issues were an integral part of our research. We found that environmental and contextual factors coupled with global distribution can impact factors such as access, cost, and cultural and linguistic diversity, and, in turn, influence the choice of methods. Revealing our focus on privacy presented cultural and linguistic diversity, and, in turn, influence the choice of methods. Revealing our focus on privacy presented the risk of priming our informants. Therefore, we advertised the goal of the research as a “study of collaborative work practices.” In addition to avoiding biased responses, this larger scope of looking broadly at all work practices of the project members led to the important side benefit of uncovering contextual links that enabled a deeper understanding of our specific research questions. Specifically, we used four methods to study the project: (i) non-participant observation of meetings, (ii) field visits to several of the project sites, (iii) semi-structured interviews with individual project members, and (iv) a face-to-face meeting with the project manager. Despite the care we took, achieving this was not always fully successful. For instance, one of our interviewees interpreted some of our questions as driven by an agenda of installing “monitoring” technologies at the workplace, and after the interview informed his manager that he would quit if this were the case. Similarly, one project member chose not to complete the online questionnaire because she was “offended” by the questions.

1 Despite the care we took, achieving this was not always fully successful. For instance, one of our interviewees (mis)interpreted some of our questions as driven by an agenda of installing “monitoring” technologies at the workplace, and after the interview informed his manager that he would quit if this were the case. Similarly, one project member chose not to complete the online questionnaire because she was “offended” by the questions.
participants, and (iv) an online questionnaire. Details of our methodology along with methodological insights resulting from our practical experience of applying these methods when studying Project X are discussed in-depth in Patil et al. [2].

3. ISSUES AND CHALLENGES

Although we learned a great deal about Project X using a long-term, multiple-methods approach, we faced issues and challenges in including two commonly used qualitative techniques that often provide rich insights regarding a setting and associated individuals: participant observation and shadowing. The considerations that ultimately led to our decision of not using these techniques in our methodology can serve to seed discussions regarding how to tackle these issues and possibly mitigate or overcome the challenges they present when studying distributed software development projects.

3.1 Participant Observation

Participant observation would have allowed us to gain first-hand experience with the work of those whom we were studying and encounter the aspects of work practices we were interested unpacking. However, we were faced with several hurdles:

- Software development is a skill that requires years of training and practice in the profession. Luckily, most members of our team did have experience in writing software programs. However, in order to be a member of the specific project under study, one also needs project-specific skills and knowledge beyond general professional experience. The learning curve for being a competent participant would have been prohibitively time-consuming and expensive despite our professional software development experience.

- A second challenge in this regard is that gaining access as a participant requires that other project members find the researcher as competent as any “normal” project contributor would have been. This is particularly true when the other project members are not aware of the researcher role of the participant observer. Moreover, getting hired to work on a software development project (in any role) is typically a competitive and elaborate process. A participant observer needs to gain access as a project member in such a way that does not overtly sidestep this process. These risks, that could lead to the legitimacy of the observer’s role as a project contributor being called into question, complicated participant observation of software development project.

- Distributed software development projects can be quite large with a variety of contributor roles, such as developer, tester, systems engineer, software architect, manager, etc. It is often not clear which of these roles could be the most beneficial for the participant observer. If multiple roles are equally important, then the observer may need to switch roles or more than one participant observer will be needed. Geographical distribution also presents the complication of deciding the physical location of the observer from among the various sites of the project.

- The complex nature of distributed software projects can lead to individual participants having only a small, local view of the whole project. However, deeper understanding of the research question also requires a global perspective on how the various components fit together. Such a perspective may not be easily obtainable with a single participant observer.

- The ethical issues regarding disclosing the dual role of the participant observer to others in the setting is compounded when the topic of investigation is privacy. Firstly, in this case, the topic necessitates that the observer’s researcher role be unknown to the informants. Secondly, in the participant role, the observer may be trusted by the informants with privacy-sensitive matters. While these situations provide rich research insight, they may also raise issues of breaching trust and expectations of confidentiality.

There are a few possibilities to address the above factors. For instance, one of the current project contributors could be recruited to serve as a participant observer. While this overcomes the issue of access and legitimacy as a project contributor, the individual may need significant training in research methods, thus adding to the time and expenses. In our case, due to the demanding and time-consuming nature of Project X, it was not possible to recruit a project member willing to take on this additional commitment. The issues of multiple project roles and locations could be overcome with multiple participant observers. However, this not only results in significant increase in research costs, it also multiplies the above-mentioned issues of gaining legitimate access as a participant.

3.2 Shadowing

Apart from non-participant observation of management meetings, we did not observe people individually while they were engaged in their work on Project X. For starters, such shadowing, which is often useful for analyzing colocated, synchronous, highly-coupled collaborative practices, may not be as effective when studying distributed collaboration because a majority of the collaborative activities are loosely-coupled and computer-mediated and therefore cannot be observed directly. Instead, it might be more illuminating to study project outputs and/or usage logs of collaborative tools. Moreover, in such cases, shadowing captures mainly the interaction of the shadowed individual with the technology rather than with the party with whom the individual interacts. This leads to a one-sided view of the interactions because the other party to the technology-mediated interaction cannot be observed. Yet, in-situ observation, especially when coupled with the analysis of logs of computing activities, might have helped uncover discrepancies between participant responses and actual behavior. However, we decided against shadowing since it might have made study participants concerned about employer surveillance. Further, shadowing would have been possible only at the work places because it was impractical to observe people while they were telecommuting from home. In our specific case, shadowing also presented an additional challenge due to our focus on privacy. We may not have been able to obtain permission from privacy-sensitive individuals or for observing an individual when he or she was dealing with privacy-sensitive matters. Yet, these were the individuals and situations with
the largest potential for insight into our research topics.

4. CONCLUSION

Geographically distributed software development is one of the most common forms of distributed collaboration. Our experience of studying a global corporate software project highlighted several methodological issues and challenges for using participant observation and shadowing to study such projects. Discussions and further work on addressing these aspects can be of benefit to the HCI, CSCW, and software engineering communities. Most open-source software projects are also globally distributed. However, they operate with fundamentally different philosophy, incentives, motivations, and goals compared with corporate software projects. While these differences do limit the applicability of our discussion to studies of globally distributed corporate software projects, extension of the discussion to open-source projects is also a possible avenue for future exploration.

5. ACKNOWLEDGMENTS

The research described in this paper has been conducted in collaboration with Alfred Kobsa, Ajita John, Doree Seligmann, and Lynne S. Brotman. The research described has been supported by NSF grant nos. 0205724 and 0808783.

6. REFERENCES

