Empir Software Eng (2014) 19:822–856
DOI 10.1007/s10664-012-9240-x

Data collection in global software engineering research: learning from past experience

Rafael Prikladnicki · Alexander Boden · Gabriela Avram · Cleidson R. B. de Souza · Volker Wulf

Published online: 16 January 2013

Abstract Global Software Engineering has become a standard in today's software industry. Research in distributed software development poses severe challenges that are due to the spatial and temporal distribution of the actors, as well as to language, intercultural and organizational aspects. These challenges occur in addition to "traditional" challenges of the domain itself in large-scale software projects, like coordination and communication issues, requirements volatily, lack of domain knowledge, among others. While several authors have reported empirical studies of global software development projects, the methodological difficulties and challenges of this type of studies have not been sufficiently discussed. In this paper, we share our experiences of collecting and analysing qualitative data in the context of Global Software Engineering projects. We discuss strategies for gaining access to field sites, building trust and documenting distributed and complex work practices in the context of several research projects we have conducted in the past 9 years. The experiences described in this paper illustrate the need to deal with fundamental problems, such as

Communicated by: Premkumar Thomas Devanbu

R. Prikladnicki (✉)
PUCRS, Porto Alegre, Brazil
e-mail: rafaelp@pucrs.br

A. Boden · V. Wulf
University of Siegen, Siegen, Germany

A. Boden
e-mail: alexander.boden@uni-siegen.de

V. Wulf
e-mail: volker.wulf@uni-siegen.de

G. Avram
University of Limerick, Limerick, Ireland
e-mail: gabriela.avram@ul.ie

C. R. B. de Souza
Vale Institute of Technology and Universidade Federal do Pará, Belém, Brazil
e-mail: cleidson.desouza@acm.org

understanding local languages and different cultures, observing synchronous interaction, or dealing with barriers imposed by political conflicts between the sites. Based on our findings, we discuss some practical implications and strategies that can be used by other researchers and provide some recommendations for future research in methodological aspects of Global Software Engineering.

Keywords Global software engineering · Research methods · Qualitative research · Data collection

1 Introduction

Over the last 15 years, the Software Engineering (SE) community has witnessed considerable developments in a relatively recent area of research, named by some Distributed Software Development (DSD) and by others Global Software Engineering (GSE) (Carmel 1999; Herbsleb 2007). No matter the terminology used, nowadays, companies are increasingly locating their software development processes in various regions across the globe, taking advantage of technology improvements and global trade regimes (Herbsleb and Moitra 2001; Damian and Moitra 2006). These companies aim at competitive advantages associated with cost, quality, and flexibility of software development, in a way that is expected to increase productivity, and decrease certain risks (Sengupta et al. 2006). At the same time, the increasing spread and success of Open Source software, which is usually developed in a distributed manner, led to a different understanding of software development methodologies, as well as of new distribution and participation models. Distributed software development projects impose new patterns of cooperation among organizational entities and human actors, which cross physical, temporal, linguistic, cultural, and geographical boundaries. Because the practices of these distributed software processes are not yet fully understood and continue to evolve, empirical studies are central for the foundation of GSE as a research field. However, these studies face a variety of methodological challenges generated mainly by the distributed nature of the practices to be investigated.

In the GSE literature, empirical studies adopt different research methods, such as exploratory case studies (Prikladnicki et al. 2003, 2007; Boden et al. 2007), confirmatory case studies (Carmel 2006), surveys (Herbsleb et al. 2001), ethnographies (De Souza and Redmiles 2011; Avram 2007a), and experiments (Espinosa et al. 2007). While it is generally agreed that studying global software projects can be challenging, only a few studies discuss the related methodological difficulties explicitly. For instance, Patil et al. (2011) discuss the reasons for choosing particular methods and the experience gained from using these methods when studying a global corporate software project.

In the following, we will illustrate the methodological challenges of GSE studies and some strategies for dealing with them. In doing so, we will first examine some general challenges we identified in the literature on qualitative research in Software Engineering, and then we will discuss these challenges against our own experiences in conducting research in the specific context of Global Software Engineering projects. The challenges that we present in this paper are not unique to qualitative studies, as quantitative studies also require collection of data from the field (Tosun et al. 2008). Moreover, some of the challenges and recommendations described are also present in collocated projects. However, we argue that in GSE they are aggravated for different reasons (technical, social, political, etc) that will be discussed in this paper. Apart of that, we will also report and discuss novel challenges that we identified based on the distributed nature of global software projects.

The rest of this paper is organized as follows. Section 2 presents a brief theoretical background for the research methods discussed in this paper. In Section 3, we present the challenges we identified in the literature on Qualitative Software Engineering research, as well as on conducting ethnographic investigations in networks and virtual settings. In Section 4, we present an extension of the challenges presented in Section 3, based on our own experiences from GSE projects. We also present how we dealt with the identified challenges. Section 5 presents a discussion about the main challenges identified. Section 6 presents the threats to credibility and finally, Section 7 concludes the paper.

2 Theoretical Background

Software Engineering is a discipline highly dependent on people and the environment where it takes place (Dawson et al. 2003; De Souza et al. 2009), and for this reason some researchers favour an interpretivist approach (Carver et al. 2004; Dittrich et al. 2007). In this research paradigm, all research has to be interpreted within the context in which it takes place and even the researcher must be considered part of this context. This can be contrasted with quantitative approaches, like experiments, which, ideally, can provide results that are generalizable, i.e., context-independent (Creswell 2003). When selecting an empirical method for a GSE study, a number of factors must be taken into account: the theoretical stance behind the methods, the practical considerations in the application of the methods and last but not least, the research questions one wishes to answer (Easterbrook et al. 2007). Before discussing qualitative research in more detail, it is important to set a common ground by briefly discussing the nature of software development in general.

2.1 The Nature of Software Development Work

A fundamental characteristic of many software systems is their size and complexity, which puts them far beyond the ability of any individual or small group to create or even to understand in detail (Hine 2000; Curtis et al. 1988). In other words, *software development is typically a collaborative endeavour* (De Souza et al. 2009) where teams of software engineers work together to achieve a common goal: the delivery of a software system on time, on budget and according to the specification. Team sizes can vary from five to more than one thousand developers working on the same project and their work has to be managed and coordinated efficiently to avoid delays, extra costs, conflicting and duplicated work, etc. Complicating things further, these developers might be located in different sites around the globe. This creates the need to deal with cultural issues and also reduces the chances for informal communication, making the coordination of their work more difficult (Herbsleb et al. 2001; Kraut et al. 1990).

In addition, *software development is an artifact-based task*: several different artifacts are produced during the construction of the software, in addition to the source-code (Kraut and Streeter 1995). These artifacts are used as “shared representations” through which developers can communicate with each other and coordinate their activities. Some examples of artifacts are: requirements specifications, test cases, analysis and design diagrams, database tables, user interface sketches, product backlogs, scenarios of use, inspection notes, requests of changes, project management plans, and so on. In general, three types of artifacts are involved: unstructured text (minutes of meetings, requirements specifications), structured text (inspection notes and requests of changes), and diagrams (entity-relationship, data-flow, class models, etc).

The types of artifacts and the moments when they are created are specified by the software development process adopted in the project (e.g., the Unified Process (Jacobson et al. 1999)). Furthermore, *these artifacts are not independent of each other*: there are relationships among them such as dependency, adequacy and evolution, among others (Spanoudakis and Zisman 2004). One of the most interesting and important relationships is the *dependency* relationship. It means that an artifact *A* may depend on another artifact *B*, and as a consequence, whenever *B* is changed, *A* must be changed too or, at least, reviewed to ensure the consistency between these artifacts. This relationship is important because it affects the coordination of software development tasks: in this case, the developers responsible for A need to be informed about the changes in B. The developers responsible for artifacts A and B may even need to coordinate their work to guarantee the consistency between their artifacts. In a geographically distributed context this coordination may be problematic (Herbsleb and Moitra 2001; Herbsleb et al. 2001; De Souza et al. 2007).

In general, distributing software development activities among several teams who work in different organizations, countries and time zones increases the complexity of their tasks and decreases the developers' opportunities for formal and informal communication, making it much harder to work efficiently (Carmel 1999, 2006; Herbsleb and Moitra 2001; Herbsleb et al. 2001; Herbsleb 2005). We will describe the problems of distributed and global software development in the next section.

2.2 The Challenges of Global Software Engineering Research and Practice

In 2001, Herbsleb and Moitra (2001) summarized the main problems of GSE as a set of dimensions: strategic issues, cultural issues, inadequate communication, knowledge management, project and process management issues, and technical issues. In their view, the GSE research agenda had to include empirical studies to investigate these problems so that the software engineering research community could gather a better understanding of them. Four years later, Herbsleb (2005) argued that computer science is necessary, but not sufficient for understanding and overcoming the problems we face in software engineering. His observation was based on the fact that we need to understand not only the properties of the software itself, but also the competencies and limitations that humans bring to software development projects. In other words, we need to understand the human capabilities in order to determine how to enhance them in the context of software engineering; for this reason, the author proposed to start creating a culture of interdisciplinary research. Of course, other researchers have already begun taking human aspects into account in the study of software development (Perry et al. 1994; Sharp and Robinson 2004; Plonka et al. 2012).

In 2006, Damian and Moitra (2006) affirmed that a GSE body of knowledge had been created over time, and that the discipline had grown through a practice-influencing research, although there was "still a significant understanding to be achieved, methods and techniques to be developed, and practices to be evolved before it becomes a mature discipline". One year later, Herbsleb (2007) described a desired future for GSE, talking about the problems and challenges that were still there in order to achieve that vision. In his view, GSE needed the following capabilities:

- To use available resources no matter where they are;
- To plan practices and technology to support the level of coordination required among the sites;
- To achieve shared understanding of requirements;
- To measure how a software architecture fit with a given organization that will build the system; and
- To effectively manage change.

Herbsleb (2007) also pointed out several research challenges and goals for GSE as a discipline, bringing several examples of research being developed. The research challenges for GSE include software architecture, requirements elicitation and communication, environment and tools, and orchestrating global development. Several challenges are still there, and substantial part of the research on GSE has to deal with understanding not only processes and plans, but also practices and people in order to develop better tools and approaches. For this reason, interpretive methods such as qualitative methods are important. As an example, King and Torkzadeh (2008) did an assessment of 43 papers submitted to a special issue on Information Systems Offshoring of a top Information Systems journal and found that qualitative methods were the research approach used in most of the studies submitted.

2.3 Qualitative Research in Software Engineering

According to Seaman (1999), empirical studies had begun to achieve significant recognition in the Software Engineering research community in 1999. She also noted that empirical studies were addressing the human role in software development. One of the major reasons for engaging in qualitative research is to understand a given phenomenon in more depth (Taylor and Bogdan 1984). In addition, according to Creswell (2003) the motivation for qualitative research is to collect open-ended, emerging data, in order to identify themes and recurring patterns. As qualitative approaches force the researcher to delve into the complexity of the problem rather than abstract it away, the findings are usually highly informative (Wohlin et al. 2003). Qualitative methods can also be used to explore substantive areas about which little is known or about which much is known to gain novel understandings (Strauss and Corbin 1998).

Practitioners agree that software development presents a number of unique management and organizational issues that need to be addressed and solved, leading to studies related not only to Software Engineering technical issues, but also to contextual and organizational issues, as well as to topics at the intersection between the technical and the contextual. In the past few years, the amount of studies based on qualitative approaches and focusing on human aspects in Software Engineering has grown constantly, many of them being published in prestigious Software Engineering venues that gradually opened up to accepting studies based on a qualitative perspective.

When compared to quantitative studies, findings from qualitative studies are often hard to generalize and transfer to other settings (see (Wulf et al. 2011)). However, especially when complex Software Engineering issues are concerned, qualitative methods are very valuable in complementing quantitative approaches, which sometimes have difficulties in contextualizing their findings (Paré 2007; Dybå et al. 2011). The reason is that when human subjects are involved, the size of samples is usually small, because controlled experiments can be very expensive to run (Biolchini et al. 2005). Alternatively, qualitative studies can generate well-grounded hypotheses and findings that incorporate the complexity of the phenomenon under study (which can then in turn be tested using quantitative methods). They also offer explanations for quantitative results and can hint to new areas for future study. Furthermore, they are appropriate when variables are not defined or quantified and there is little prior theoretical or empirical work on a specific topic (Klein and Myers 1999).

2.4 Challenges of Conducting Qualitative Research in (Global) Software Engineering

For a long time, qualitative research has traditionally been focused on localities, i.e. the field of observation was usually a (local) community, organization, or cultural phenomenon in a local or regional context. However, globalization, in terms of easily available communication technologies and cheap means of traveling, has blurred the notion of "field" and raised new challenges for qualitative research. This new situation has an important impact on the theoretical stance of ethnographic research, as the unit of analysis itself—and the boundaries of the field site—may have to be redefined in order to match the concrete situation. Hence, instead of focusing on "the field" as a locality, the researcher needs to take into account multiple socio-political sites and locations that are interconnected (Brown et al. 2007; Gupta and Ferguson 1997). This new understanding of ethnography resulted in a spread of postmodern approaches that shift the focus towards understanding the field as a relation of multiple sites. For example, ethnographer George Marcus developed the concept of multi-sited ethnography that requires the ethnographer to take into account the different fields (in terms of social contexts) in which people live (and work) (Marcus 1998), and investigate their interconnections, overlaps, and relations. Other scholars have suggested an analytical focus on the mobility of actors and the ways in which they produce spaces (which are understood as social constructs rather than physical places) in their interaction (Dourish 2006). Such approaches often argue for *mobile* ethnographies instead of multi-sited ones (Hine 2000), thus indicating that "the field" has lost its relevance as a focus for research.

The complexity that qualitative methods have to face in the context of such interconnected fields also applies to cases of Global Software Engineering, where multiple companies or teams located in several countries are working to achieve a shared goal. Given the large number of people involved in such projects, all playing different roles and having different expectations, the many artefacts produced during any software development project and the way these artefacts are interconnected, managing (and understanding) them is a very complex task (see Section 2.1). More importantly, to conduct research in such an intricate context like global software projects is in many ways as challenging as the very practice of software development. The large number of developers playing different roles means that a researcher must observe or interview several different informants; the large number of interconnected artefacts creates the need for the researcher to be able to find, access, and understand a whole range of artefacts, often outdated or incomplete. Furthermore, in order to successfully understand a software development project, a researcher also needs to understand the "tools of the trade" (Geertz 1973) employed by the developers involved in the study so that (s)he would be able to make sense of what is going on in the field. On top of this, tools employed can be different from one site to another.

This all means that conducting qualitative research in GSE can be time-consuming and demanding for the researcher, requiring the handling of social relations in sometimes unfamiliar settings, dealing with ethical aspects, managing and interpreting often ambivalent narrative data (Oates 2006). For example, one important challenge that has been discussed in this regard is the time budget aspect. As the researcher has to visit several sites, (s)he has to divide the available time in order to spend time at each site. This can have implication for the ethnographic inquiry, which needs personal immersion and a considerable amount of time in order for the researcher to obtain an understanding of the sites (and of the inter-site connections) under study (Wittel 2000). This understanding can be quite complex and difficult to grasp (Geertz 1973). Furthermore, there is no simple set of rules to follow, and a lot of relevant factors cannot be controlled (Parnas 2003). Having to deal with all these aspects make qualitative research in Global Software Engineering very challenging.

Given the complexity of such an endeavour, only a few studies ventured to describe the challenges of conducting research in Global Software Engineering, and more importantly, the approaches that should be used to handle these challenges (Brown et al. 2007; Gupta and

Ferguson 1997). A recent issue of the Information and Software Technology Journal included a selection of papers dedicated to studying work practices in Global Software Engineering trying to address the mater (Avram and Wulf 2011). In order to address this gap, we conducted a literature review to identify known challenges and approaches used for dealing with these challenges. Initially, we focused on challenges for conducting qualitative studies on distributed teams, but it soon became clear that a broader approach was necessary. Therefore, we expanded the scope of our literature review to encompass references describing challenges in conducting qualitative Software Engineering studies. Finally, we added a list of challenges based on our experiences conducting these studies in global software development projects.

It is important to say that the authors' experiences are based on the assumption that the research team has physical access to at least one of the sites to be investigated. If only one site is studied, the challenges in collecting qualitative data are pretty much similar to any other qualitative study. This issue is discussed in Section 4.1. However, in order to get a deeper understanding of distributed work practices, we argue that it is necessary to regard Global Software Engineering as a holistic phenomenon, involving the study of several sites, which implies data collection at these sites. This second approach creates additional challenges compared to traditional qualitative data collection. While we recognize that some of these challenges are similar to challenges in single-sited qualitative research, we argue here—and will illustrate with examples—that these challenges are exacerbated in distributed settings (Section 4.2). Before detailing challenges in distributed contexts, the next section presents a review of challenges for conducting qualitative Software Engineering research resulting from our literature survey.

3 A Review of Challenges of Qualitative Software Engineering Research

The amount of studies that identify challenges in conducting research in Global Software Engineering is limited. While there are many references in the literature to the challenges of qualitative research in general, we have found very little information on the particular challenges in Software Engineering or regarding distributed and virtual settings. In this section we will provide an overview of the challenges found in the general literature on research methodology (see Fig. 1 for an overview of these challenges). The challenges presented here are also faced by Software Engineering researchers who use industry data, and many of the identified challenges also apply to global software projects. Some of these challenges also apply to quantitative studies, and we will indicate that as appropriate.

In the remainder of this section, we will describe the "practical problems" faced by researchers in relation to each reported challenge. For instance, one challenge is to gain access to companies, which means practical problems like negotiating with companies the participation in the qualitative study, legitimizing the study, etc.

3.1 General Challenges Reflected in the Literature on Qualitative Data Collection in Software Engineering

3.1.1 Gaining Access to Companies

Qualitative studies require data to be collected "in the field" in order to capture the context in which work is performed. For commercial software development projects, it is necessary to identify companies that are willing to participate in a study, establish contact with them and

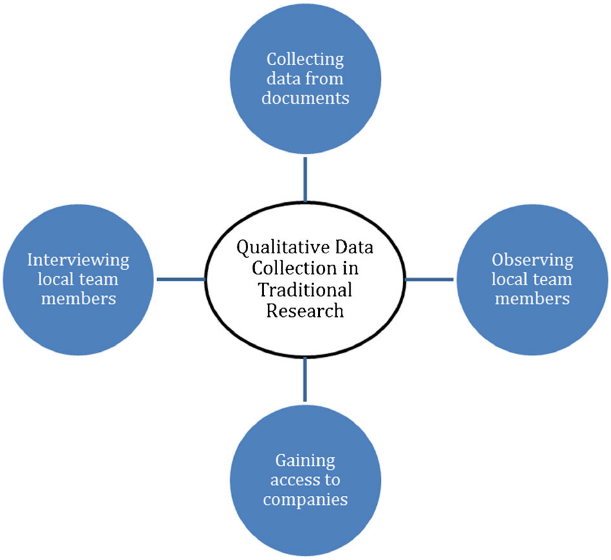

Collecting data from documents
Interviewing local team members
Qualitative Data Collection in Traditional Research
Observing local team members
Gaining access to companies

Fig. 1 Challenges of collecting qualitative data in Software Engineering research

convince them to participate in the study. Convincing a company to participate in a study can be, in many cases, challenging, requiring thorough explanation of the goals of the research and of the methods to be deployed, as well as providing reassurance to preserve the anonymity of the participants and of the company, if required. *In short, the first practical problem is to convincing companies to participate in qualitative studies.*

In business contexts, another problem is the legitimization of the study. As ethnographic methods require time, and often the data collected might not have an obvious use for the company under study, the benefits might be questioned. The researcher has to give convincing explanations regarding the reason for requiring access to the site. As access is often negotiated, from a business perspective, getting access may depend on what the ethnographer has to offer in terms of expected benefits for the company under study (Wittel 2000). In consequence, the researchers have to familiarize themselves with the company's perspective and try to understand what would be relevant for the practitioners, in order to be able to negotiate and persuade the companies to grant them access.

As we will discuss in more details in Section 3.1.4, the researchers are commonly not given access to all the artefacts in an organization. Usually they rely on the goodwill of informants, who share what they believe would be relevant for the researchers (Randall et al. 2007). Even if the researchers are granted access to artefacts, some areas of activity might be regarded as 'forbidden' and off limits for the observers. In any organization, there are likely to be gatekeepers who can deny or grant access to these areas. Especially in large organizations and in studies covering several departments, *dealing with gatekeepers is an ongoing practical problem*. This can also have political implications, in the case of ongoing conflicts inside the organization (Randall et al. 2007). For instance, sponsorship and association with particular vested interests can sometimes create problems (Randall et al. 2007). The majority of organizations have a complex structure, and sometimes it can be difficult to find someone to take responsibility for granting access. It should be noted that the first two problems (to convince companies to participate in the study and to legitimize this study) are also common problems in quantitative studies.

3.1.2 Interviewing Local Team Members

One of the practical problems of interviewing local team members include the fact that *interviews provide "indirect" information* that is filtered through the views of interviewees. Informants' abilities to perceive and articulate information sought by researchers differ substantially, and the information is usually provided in a designated location rather than in the natural field setting (Creswell 2003). Lethbridge (2005) argues that interviews and questionnaires are the most straightforward instruments, but the data they produce typically present an incomplete picture. Conducting research in the "field" is often stressful for the informants; they are more likely to be willing to participate if they feel comfortable with the researchers and feel they are partners in studies focusing on issues that they consider important. Creswell (2003) considers that interviews are useful only when participants cannot be observed directly, or if they can provide historical information.

As for *the selection of local team members to be interviewed*, most studies in Software Engineering adopt what is called 'convenience sampling', meaning that the researcher involves whoever is available and willing to participate. This may result in various types of bias, such as self-selection bias (those most interested in the study may have specific characteristics). In this case, it is important that the researcher gathers other types of data (records, manuals, documents, etc) in order to triangulate the data provided by the interviewees.

Finally, there are many ways to conduct interviews. Telephone interviews are not as personal as face-to-face interviews, yet they still provide researchers with opportunities to clarify answers to questions and further probe interesting responses (Creswell 2003). Although this technique is popular in opinion polling and market research, it is little used in empirical Software Engineering (Lethbridge et al. 2005). In face-to-face or telephone interviews, the researchers have to *schedule meetings with the respondents* which can prove problematic given software engineers' busy schedules (Lethbridge et al. 2005). Oftentimes meetings are cancelled or re-scheduled, therefore the researcher needs to be flexible, patient, and deal gently with last-minute changes in the meetings' schedule.

3.1.3 Observing Local Team Members

The main practical challenge of observing local team members is the need to *interpret the observations*. As observed team members may see the study as intrusive (Creswell 2003), participants may adjust their behavior as they feel observed, thus altering the observed situation, especially at the beginning of the observation period. After a while, this adjustment of behavior tends to disappear or to be less common, but it still is an issue. Researchers have to keep in mind that they observe people, who in turn are observing the researchers (Gobo 2008). At least, this has to be taken into account during the analysis of the field material, where the ethnographers have to reflect on their own impact on the observed situations.

Another practical issue that is connected to the main challenge is the need to *continuously reflect on one's own perspective*. Researchers who become too involved may lose perspective on the phenomenon being observed. Sometimes it can be better to be less familiar with the observed field, as it allows researchers to notice aspects that otherwise may remain hidden. At the same time, researchers have to be knowledgeable enough to make sense of the observed practices and situations, otherwise their field notes tend to be too generic to be useful for analysis. In addition, longer data collection periods might be required for the researcher to acquire enough knowledge to understand the observed situations. In summary, the researcher needs to be continuously questioning the input provided by the informants as well as her\his own assumptions about what is being observed.

Last but not least, it is an important practical challenge to *keep in mind the limitations of observation*. Especially with regard to software development practices, the aspects that can actually be observed are limited (Seaman 1999) (as parts of the work are done in isolation in front of the screen). As software developers reveal their thought processes most naturally when communicating with other software developers, it can be a good approach to look for incidents by studying the communication (Seaman 1999). However, as such incidents are hard to predict or plan ahead, chances are high to miss important situations, especially when one researcher has to observe a whole team or company.

3.1.4 Collecting Data from Documents

Software development relies on various types of electronic documents which are often highly interwoven with the work practices in which they are created and used. However, as some documents (such as the source code) are often considered as being highly confidential, access to digital repositories such as the version control system or even the bug tracking system is often restricted or only partially granted. Hence, one important challenge of working with documents is *dealing with the limited access* to certain kinds of documents, which can make it more difficult to get a clear picture of what is going on. This challenge is also quite common in quantitative studies.

Furthermore, even when full access to digital repositories is granted, researchers should be aware that not every relevant document is accessible through online repositories. *Finding important offline documents* (such as hand written notes and scribbles) and incorporating them into the analysis can be challenging, especially as these are sometimes considered private and not exchanged openly between team members.

Last but not least, there is the challenge of *capturing the context of artifacts*, which need to be regarded as limited and simplified representations of the events that led to their creation, and not as "objective" proof for those events (for example, bug reports in bug tracking systems can differ significantly from how the bug actually was discovered, discussed and fixed (Aranda and Venolia 2009)) At the same time, artifacts can be incomplete or some documents can be inaccurate or altered for various reasons. In other words, the documents need to be understood and analyzed in the context of the practices that have produced them. Understanding how these documents were created, used and perhaps even stored and digitized can be very important when conducting qualitative research, and creates the need to make several visits to the sites and spend a lot of effort to understand the relevant practices in detail.

3.2 Summarizing the Challenges and Problems for Conducting Qualitative SE Research

A summary of the different challenges and problems associated with qualitative research with software development teams is presented on Table 1 below.

4 Extending the Previous Challenges to Distributed Settings—An Experience Report

In this section, we will compare our experiences in conducting GSE research with the challenges we have identified in Section 3. Our experiences are based on several projects conducted in the past years (please see the Appendix for a detailed description of each of the projects). Collectively, the authors have performed 6 qualitative studies involving several globally distributed teams from 11 different countries. Then, we will discuss our experiences

Table 1 Challenges and problems with qualitative research of software development teams

Challenge	Practical problem
Gaining access to companies	Negotiating with the companies and all sites involved
	Legitimizing the study
	Dealing with gatekeepers
Interviewing local team members	Dealing with "indirect" information
	Selecting local team members to be interviewed
	Scheduling interviews
Observing local team members	Interpret the observations
	Reflect the own perspective
	Keep in mind the limitations of the observations
Collecting data from documents	Deal with limited access to documents
	Find offline-documents
	Capture the context of the artifacts

in dealing with the challenges from the literature. Last but not least, we will present and discuss additional challenges we have found (see Fig. 2 for an overview—light gray circles indicate additional challenges).

4.1 Our Experiences with the Already Identified Challenges

4.1.1 Gaining Access to Companies

In all our studies, *one of the problems was to find companies with distributed projects willing to participate*. One of the reasons for this difficulty is that *distributed development can be a*

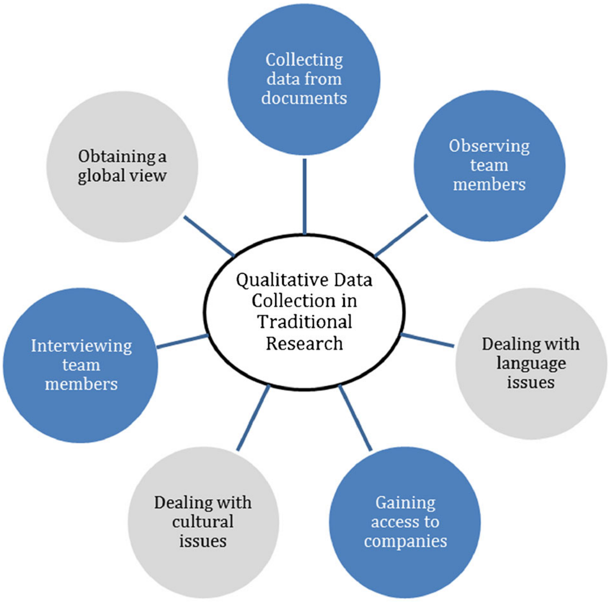

Collecting data from documents
Obtaining a global view
Observing team members
Qualitative Data Collection in Traditional Research
Interviewing team members
Dealing with language issues
Dealing with cultural issues
Gaining access to companies

Fig. 2 Challenges of collecting qualitative data in GSE research

controversial issue amongst the employees of a company, so some companies might be hesitant to participate. For instance, during the acquisition phase for Boden et al.'s study (2007), acquiring partner companies at the CeBIT in Germany turned out to be very inefficient, since only few of the present companies advertised their offshoring endeavours (probably due to the critical discussion of job losses through to offshoring in the mainstream media at that point of time). Furthermore, some companies that were identified as possible sites for the research refused to take part in the study, apparently assuming that they would get bad publicity just by participating in this study. This problem was probably aggravated by the fact that these companies were never involved in qualitative research before and could not estimate the consequences of participating in such a project.

Another Problem is Dealing with the Possible Frictions and Rivalries that may Go Along with GSE Projects As GSE (i.e. offshoring) may result in job losses and shifts of competencies and power within local teams, it is necessary to clearly define the role of the researcher in the context of the project. For example, in Boden et al.'s study (2007), one of the interviewees, an area manager of a German SME, offered the researchers the chance to accompany him on a journey to an offshore site, where he wanted to negotiate a new offshoring project. However, the researchers declined the opportunity, due to financial limitations and doubts concerning the relevance of this specific meeting for the research question that focused on operational rather than strategic aspects of GSE. Thus, the researchers preferred to schedule the participant observation to a later stage of the project, when software development was scheduled to take place. This decision led to unforeseen frictions between the research team and the management of the company. During the discussions that followed, it became clear that the area manager wanted to be accompanied by the researcher in order to be able to justify his decisions to his superiors. As the decision to extend the offshoring agreement was controversial in the company, he expected to gain support for his arguments by involving a research team in the planning process. The refusal to provide this type of support led to frustration from his part, and he declined to work with the research team any further. Thus, it is necessary to clearly discuss the expectations and aims of all partners involved (researchers, management, developers), in order to be able to deal with these possible frictions and conflicts of interest, and to avoid getting involved in the social/micro-political conflicts that arise around some projects (Argyris et al. 1985). However, since permission is needed to gain access to the field, researchers often cannot avoid getting involved in political quarrels on site (Randall et al. 2007). Although this is a general problem of ethnographic field studies (and even some quantitative studies), geographic distribution and political tensions related to offshoring can make this challenge much more difficult to deal with.

In "traditional" qualitative studies (including ethnographic ones), access is usually facilitated by so-called gatekeepers (see Table 1). In empirical studies of distributed projects, the ethnographer cannot expect to automatically get access to all development sites after establishing access to one of them. Hence, in the worst-case scenario, it might be necessary to get the support of as many gatekeepers as there are sites to be studied. Again, this may take time, allowing less time for the data collection and analysis itself later on (Wittel 2000).

Another *problem* we faced was related to *the legal aspects of gaining access to distributed sites*. We found this to be very problematic, because different sites have different regulations due to local laws. In this case, our experiences were diverse: in some cases, this was not problematic at all (De Souza and Redmiles 2011); in others it demanded additional effort. For instance, if the company in question already had agreed procedures (e.g., for granting restricted access to the intranet to researchers), it was just a matter of filling up the right forms and waiting for approval (Avram 2007b; Avram et al. 2009). However, when the company had never encountered a similar situation before, it was difficult and time consuming to obtain access to the internal mailing lists, databases and documents, requiring a great deal of persuasion and explanation: in Boden et al.'s study (Boden et al. 2007), one partner company feared that granting access to internal databases and documentation to researchers might violate legal agreements with their customers. Hence, they wanted the researchers to sign a standard non-disclosure agreement, the same they had agreed to with their customers. Unfortunately, it was entirely unclear how this would have affected the researchers' possibilities to publish the (anonymized) findings. It took a lot of negotiation (involving professional advice from a lawyer) in order to come to an individual agreement in such a legally complex situation without endangering the research.

In general, in our experience a research proposal has to be developed and presented to different stakeholders in order to get permission to collect data. In one of the studies, face-to-face meetings had to be scheduled with contact persons in four of the five companies, in order to get the proposal approved (Prikladnicki et al. 2007). To be more specific, we included in the proposal the following information: the reasons why the company was chosen for the study, the activities planned, how the results would be reported, and the company's involvement.

4.1.2 Interviewing Team Members in GSE

As far as interviews with local team members are concerned, problems similar to those of most qualitative research projects apply (see Section 3.1.2). However, some of the problems of interviewing people face-to-face can also be exacerbated by distance and technology mediation. For instance, *getting the consent of remote people for an interview before visiting their site can be difficult.* Software developers are usually extremely busy and under a lot of pressure, so they may be easily annoyed by requiring a slot of their time and preventing them from doing their actual work, especially if the request for an interview is not properly prepared. At the same time, trips to remote sites are costly and possibilities for travelling can be limited. As a result, the displacement between researcher and the field can create problems, as the dependency on written forms of communication makes access to the field less personal, and less direct. As the interaction between the practitioners is often limited to words and a couple of emoticons, misunderstandings and misinterpretations are more likely to occur. At the same time, scheduling interviews with remote team members is a challenge in itself: motivating a person the researcher has never met to confer her\his time for an interview might prove extremely difficult.

To make things worse, in our experience, distance and technology mediation make it very easy for the informant to decline or avoid being interviewed. From what we have witnessed, it helps a lot to ask a local (internal) liaison to forward the request either directly to the potential interviewee, or to a remote liaison who can then serve as a proxy, as illustrated by Prikladnicki et al. (Prikladnicki et al. 2008). *In scheduling the interviews*, the researcher has to demonstrate a lot of flexibility and adjust her\his schedule to that of the remote site; finding appropriate time slots might be a challenge in itself, because people tend to claim to be always busy. In many situations, obtaining an appropriate date and timeslot for the interview is done more effectively using synchronous communication (phone, Skype, instant messaging) or by gaining access to and using the internal calendaring system of the company (De Souza and Redmiles 2011; Avram 2007a, b; Avram et al. 2009).

4.1.3 Observing Team Members in GSE

In addition to the known challenges of observing software development work, the geographical distribution can impose new challenges on the researcher. In Global Software Engineering, interaction takes place via communication media and coordinative tools (such as bug defect databases and shared repositories. Hence, one practical challenge is to analyze distributed development work in detail, due to the diversification of the related practices based on various tools, media and artifacts. Researchers have to take care not to ignore important channels of information while concentrating solely on the most obvious collaboration tools. Hence, observations have to be documented with great accuracy, or important aspects might be easily missed.

It is important to mention that there is a body of literature suggesting that software developers spend a great deal of their time with communication and coordination efforts (Perry et al. 1994; Ko et al. 2007; Gonçalves et al. 2011). In other words, distributed development is not done merely in front of a screen, with long periods of pure coding, except for some few unpredictable phases of intense interaction with the remote team. Software development is mostly a collaborative activity. Hence, another practical challenge is to observe these phases of direct interaction inter-teams because although they occur often, they are hard to predict. In this regard, Rönkkö (2000) reported that the most important challenge they faced was the unpredictable character of the work they wanted to study. There were several telephone calls, e-mail communications, and arranged meetings that occurred spontaneously in different locations spontaneously. Hence, due to the nature of distributed work, the observable parts of the software development practices can be quite varied in different phases of a project, and researchers need to rely to a greater extent on other sources of information—a fact that certainly has methodological implications for qualitative research.

Concentrating solely on the central actors involved in the cooperation while overlooking peripheral team members with only sparse contacts to the remote site is also dangerous. It is not always obvious who is involved in the cooperation, and *keeping track of all involved team members* can be quite challenging, especially given the other practical challenges described above.

Finally, in Boden et al.'s study (2007), one problem faced by the researchers was that *the informants often forgot to involve the researcher when they contacted the remote team*, as they did not always plan for such incidents, and as these were usually related to emerging problems which needed their attention (more than the interest of the researcher). As a result, sometimes the researcher was involved too late or not at all, so that interviews had to be conducted in order to reconstruct the events he had missed.

4.1.4 Collecting Data from Documents

As Global Software Development projects often entail the use of many different tools and media to allow the distributed cooperation to happen, it is important to get access to these systems; hence, *negotiating access to the software development tools* is an important practical challenge, as more systems are likely to be in place, and as their importance is much higher in comparison to local projects.

At the same time, while distributed software development work has the advantage of producing even richer and more diverse artefacts due to the necessities of bridging geographical distribution (for example, in Boden et al's study several companies reported they introduced bug trackers only after starting their offshoring cooperation), the information available on public networks may accumulate at an alarming rate. *Processing the available raw data in an appropriate way* can thus become a very time consuming practical problem (Hine 2000).

Furthermore, due to the geographical distance between the several sites, access to archives that contain traces of the direct communication between local and remote team members (e.g., e-mails and instant messenger conversations) becomes even more important for making sense of the data stored in more formal systems such as bug-trackers and version control systems. However, since these are exchanged between team members personally and often contain private information, it can be quite challenging to *get access these personal sources of data*. De Souza and Redmiles (2011), for instance, could not get access to this information, therefore what they did was to use a "social-network survey" (Cross and Parker 2004), i.e., one in which each team member answered questions about their frequency of communication (among other aspects) with several other informants. By using this data, the authors were able to reconstruct specific aspects of the interactions among the local and remote team members.

4.2 New Challenges Arising Due to Geographical Distribution

4.2.1 Obtaining a Global View

The distributed nature of global software projects make it necessary to observe global practices from a local perspective. Since deploying several researchers to simultaneously observe work practices at the different sites is often not feasible, cooperation processes are usually observed only from one team's perspective at a time. This often results in a lack of awareness and understanding of the remote team and may have as consequence a single-sided perspective resulting from the researcher's limited perception. As the researcher is dependent on the same media and tools for communication that are used by the software developers, the study can be subject to difficulties that mirror those of the software development itself (see Section 4.2.3). To get a broader picture of the distributed software project, the perspectives of different members of the *local* team can be very useful, if they can be brought together for a discussion. As mentioned before, aligning team members' different schedules is always a problem.

However, in our experience, visiting the *remote* site personally is usually the richest and most accurate way to understand the setup of the cooperation (even if it poses additional challenges, like dealing with language and cultural issues—see below). Another possibility is to collect data when members of the remote teams visit the local site. This was the opportunity used by De Souza and Redmiles (2011), who conducted a series of interviews with the Chinese developers while these were visiting the main U.S. site. This can be very difficult to arrange though, since these visits are rare and expensive, and therefore planned in conjunction with different events (meetings, discussions, etc) taking place and consuming most of the visitors' time, making the scheduling of interviews even more complicated.

Another problem we had to deal with was related to the complex, and sometimes controversial structure of Global Software Development projects, emphasizing the importance of understanding and explicitly reflecting upon the different local and trans-local perspectives of shared work. As relying on the perspective of one team only may nurture bias that almost unnoticeably dominate the research, initiating personal contact with remote team members is necessary. The perceptions of the shared projects can differ from team to team, especially in the case of offshoring. This can relate to the measures of success, as well as to the perceived efficiency of work practices and deployed tools. For instance, in Boden et al.'s study, Russian and German developers reported different estimations concerning reasons for the failure of a shared project. As teams blamed each other, it was very interesting to analyze the various views that revealed different perceptions of what constitutes "good software development" influencing the cooperation practices (Boden et al. 2007, 2008).

Due to the characteristics of distributed software development work, a lot of interaction takes place in *virtual* networks—seen as (virtual) places where people "form and re-form a culture or set of practices and meanings" (Hine 2000). This displacement between the researcher and the field can lead to a lack of common and mutual perception, which may hinder obtaining a clear picture of what is going on between the sites (Marcus 1998). Hence, a problem to be overcome it to get a perception of the physical layout, local unwritten rules, local practices and cultural differences of a remote site without actually visiting it. For example, in our studies researchers often had to rely on the reports of local team members who had visited remote teams and were able to provide a general overview. However, in Boden et al.'s study (2007), it turned out to be very difficult to obtain precise information even concerning some hard facts like the remote team's size. Since nobody from the local team had regular contact with every member of the offshore team, and its size and structure had changed several times during the year prior to the study, perceptions of the remote team differed. Also, it was difficult to reconstruct the exact reasons and dates of the changes. A similar problem was observed in De Souza and Redmiles' study (2011), but for a different reason. In this case, the source code written by the developers in Massachusetts (US) was evaluated by the quality assurance team located in China. In general, the U.S. developers only interacted with the Chinese developers by adding information (mostly source-code) to a shared repository. During the interviews, it was clear to the researchers that the U.S. developers did not know who were the Chinese counterparts testing their code. In general, these experiences suggest that it can be very hard to obtain a global view of what is "going on" on the remote sites by solely relying to interviews with the local team members.

Thus, apart from using e-mail and instant messaging systems for direct communication, this implies visiting and observing all involved teams during the research—if possible. If that is not possible, a researcher needs to be aware of the limitation of her\his study and of associated findings.

4.2.2 Dealing with Language Issues

Another challenge that we faced in our studies is related to the use of various languages in international teams. Although English is usually the common language used for the global team communication, it may be difficult to follow local communication, as some team members may tend to use their native tongue when they talk to each other. For instance, in Boden et al.'s study on cooperation between German and Russian software development teams, it turned out to be problematic to conduct observation and interviews with some of the Russian developers who did not speak English fluently. While these developers were able to use written communication and understand English texts, they were uncomfortable with direct communication for an interview. This led to an (unwanted) focus on some team members of the Russian team who were more fluent in English, while interviews with other, less fluent speakers were much harder to conduct and brought less insights as complex nuances of the cooperation were hard to communicate. Similarly, in De Souza and Redmiles's study (2011), the researchers conducted interviews when the Chinese developers were visiting the U.S. site. In this case, some interviews were very problematic, because the Chinese developers were not fluent in English. Additional time was necessary for the interviews, however these developers were on a tight schedule and were already on a site visit with several planned meetings with their U.S. colleagues. This, in general, affected some of the findings of the study that could not be validated. On the other hand, communicating in a foreign language can also have advantages for the researchers themselves, as non-native speakers are not subjected to the same expectations as native speakers (Lutz 2009).

Another issue with language was related to the difficulties Boden et al. had while observing some aspects of the development work at the Russian site (Boden et al. 2007). While Russian language was not at all used in the communication which affected the inter-site cooperation with the German company, the Russian developers often used Russian language for inter-team communication, as well as several software tools using Cyrillic characters in the interface for their work. While the researchers were told that all important communication and documentation would be in English, and as the Russians promised to explain everything they did during the on-site observations, there were still some blind spots in the observation and some aspects that were hard to grasp.

4.2.3 Dealing with Cultural Issues

Cultural issues are usually seen as a predominant factor for Global Software Engineering projects. According to interpretivist approaches, culture is seen as a reference framework, which stipulates roles and interpretations, and which is dynamically negotiated by the actors in the course of their daily work (Geertz 1973). This understanding of culture entails many different layers referring to national, professional, or religious aspects, which are seen as being intertwined in a complex, non-hierarchic way, and which can be hardly studied in isolation. It also includes many invisible aspects that cannot be studied directly, like values, beliefs, and attitudes. Hence, it is necessary to study culture by referring to its manifestations in the form of artifacts, practices, and routines, which can be quite challenging (see (Boden et al. 2009)).

While cultural differences might be obvious when traveling to remote sites and meeting people face-to-face, they might be very difficult to perceive during technology mediated interactions, where people tend to project their own environment and circumstances over the remote discussion partner. Furthermore, *the research methodology often needs to be adjusted to the cultural norms and customs of the remote site*, for example when choosing and formulating the interview questions according to the research objective (Prikladnicki et al. 2007, 2008). This is a challenge that exists in any research process, but can be considerably amplified by having to deal with people belonging to different cultures and communicating in a foreign language, making it harder to articulate nuanced opinions and understand implicit meanings (see 4.3.2).

Even when personal visits can be arranged, a general problem of studying intercultural teams is related to *the potential bias of the researcher involved*. Rather than describing attributes of a population from some neutral position outside the field under study, accounts of cultural meanings and practices are inevitably created from particular standpoints that set up the lines of comparison and contrast between the researcher and the individuals and practices described (Suchman 1987). Reflecting upon this bias, and accounting for it in the analysis can be quite challenging. The positive aspect of this challenge is that the cultural differences between sites and teams can be quite obvious for the researcher, if they differ from her\his own. At the same time, it can blind the researcher for issues related to her\his own culture, which is often taken for granted and thus the issues are harder to detect. This is especially the case in business environments, where the practices are usually aligned to pragmatic needs and professional practices. Hence, cultural differences are often referred to by addressing aspects of everyday's life like food, religion, or a general attitude toward ethic beliefs and values, but not in relation to software development practices, which are often thought of as being pragmatic and detached of beliefs or values.

However, even if there are obvious differences in the cooperation, *it can be quite difficult to discriminate the degree to which national cultures—and not an "us" versus "them" mentality—affect the explanations of human actors concerning the cultural differences between distributed teams*. For example, in Boden et al.'s study, actors reported different ways of dealing with documentation in the two teams. According to a German project manager, the Russian developers did not like to write documentation. Instead, they preferred to write "self explanatory" code and not linger with documentation, which—according to them—would be out-dated most of the time anyway. Hence, according to the German side, when the Russians developers were requested to send documentation on one particular feature, they wrote it on demand. Interestingly, in the interviews with the Russian developers, there was a different view on the role of documentation. From the perspective of the Russian team leader, the members of his team wrote much more documentation than the German team, who often ignored these tasks. Sometimes this lead to problems, like in the case of a German project manager who had simply forgotten to update the specifications with some change requests from the customer, and the Russian team worked for several weeks on features that had actually been dropped.

Although this is an extreme example, it illustrates how different organizational practices look from each perspective. It is questionable if these differences are due to the national cultures involved—they might as well be attributed to the well-known preference of developers for writing code over writing documentation. The tendency toward innovation and adding state-of-the-art details, as opposed to working on the features stipulated in contract, is the object of another well documented dispute between developers and project managers, irrespective of nationality (Gobo 2008). Hence, these differences may also be rooted in the organizational context of work. As the members of the Russian team had no direct contact with the customer, they needed clear and detailed instructions. The members of the German team on the other hand, worked under totally different circumstances and needed to keep a close connection to the customer. Hence, they preferred to work in an agile way, with requirements that were subject to ongoing negotiation and change. The perceived cultural differences may be simply patterns of explanation and interpretation—and it can be quite challenging to decompose these patterns during the research (Boden et al. 2009).

4.3 Summarizing the Challenges and Problems for Conducting Qualitative GSE Research

A summary of the problems identified in relation to qualitative research of distributed software development teams is presented on Table 2.

5 Discussion

While many of the challenges we had to deal with are common to most qualitative studies, we have shown that the field of GSE exacerbates existing problems due to the temporal, spatial, organizational and cultural distribution of the field. Problems start with the obvious fact that the researcher is dealing with *multiple* sites: now, the traditional concept of the "field" (see Section 2.4) from qualitative studies consists of different companies with sites that are often distributed over large distances, as well as of actors bridging these distances. This means that several of the problems the researcher will be facing now occur multiple

Table 2 Challenges and problems for conducting qualitative research in distributed software development

Challenge	Practical problem
Gaining access to companies	Convincing a company to participate in a qualitative study due to the controversial nature of distributed software development
	Gaining access to all sites
	Dealing with the politics of distributed development projects
	Dealing with the legal aspects of gaining access to distributed sites
Interviewing team members from remote locations	Convincing remote informants to participate in interviews
	Scheduling interviews with remote informants
Observing team members in GSE	Analyzing distributed development work in detail
	Observing phases of direct inter-team interaction
	Keeping track of all involved team members
Collecting data from documents	Negotiating access to the software development tools
	Processing the available raw data in an appropriate way
	Getting access to personal sources of data
Obtaining a global view	Lack of awareness and understanding of the remote team
	Putting local team members together to get an overview of the remote sites
	Problems similar to the ones in Section 4.2.3., i.e., observing local team members
	The complex and controversial structure of Global Software Development projects including perceptions of success, efficiency of tools, etc.
	Obtaining information about the remote site including the physical layout, local unwritten rules, local practices and cultural differences
Dealing with language issues	Observe and interview developers in a language not understood by the researcher
	Scheduling more time than usual to interview developers who are not fluent in the researcher's main language
	Observing tools used by remote developers which are not in a language understood by the researcher
Dealing with cultural issues	Collecting and analyzing data in different cultures
	Recognizing the researcher's own bias
	Recognizing how culture affects the explanations provided by the informants

times and differ from site to site. For instance, the researcher needs to deal with legal issues regarding the data collection in different ways since countries often have different laws regarding the privacy of their employees. Similarly, this same researcher needs to be able to gain the trust of the developers in different sites by, for instance, having a key informant or a liaison in each one of the sites. In other words, as software development is regarded as more difficult in distributed scenarios, so is researching software development in distributed scenarios.

In addition to dealing with the geographical distribution of the sites, researchers will also have to deal with different languages and different social, and (inter-)cultural issues. Doing interviews or observing work situations in foreign languages (e.g. Russian written language) can be very hard and requires further efforts in terms of negotiating access and establishing trust. Hence, getting consent for access to people or to private artifacts can be very hard and implies higher efforts for conducting research in distributed settings as compared to local settings. In general, the problem being described is one of building trust. Trust is, in fact, one of the general key prerequisites of qualitative (especially ethnographic) research: it is paramount for building relationships and gaining access to the field (De Souza and Redmiles 2011; De Souza et al. 2007). Trust has been characterized as a complex, multi-layered concept, which is—amongst others—related to expectations, experiences, and knowledge (e.g. Is the informant competent? Is his behavior predictable? Does he intend to be helpful? Is he opportunistic?) (Lutz 2009). This means that the challenge of gaining access to companies can be seen as a cluster of social challenges related to trust building, compulsory for gaining entry to the work setting, gaining acceptance, being able to 'hang around' and so on. Failure to establish the necessary trust between researchers and informants can create several problems during the data collection and might jeopardize the research.

In distributed teams, fostering trust can get even more difficult than in co-located settings (Herbsleb and Grinter 1999). There are several reasons for this. For instance, due to the geographical distribution and the scarcity of informal communication, it is more difficult to establish trust. Informal interactions are important for building interpersonal relationships among the distributed members of the team, and this has an impact on trust (Jarvenpaa 1998). Cultural differences among sites can also contribute to this problem, since similarity with others positively reinforces members' own identities and contributes to their openness to trust others (Jarvenpaa 1998).

Lack of trust is a well-documented problem in the literature of distributed software development. However, our focus here is not on the lack of trust among software developers involved in a project. The same reasons that lead to lack of trust *among team members* are the at the origin of the difficulty in establishing trust *among informants and researchers* involved in a distributed project. As discussed before, trust is a key prerequisite of ethnographic research that is necessary for building relationships and gaining access to the field (De Souza and Redmiles 2011; De Souza et al. 2007). However, since visits are usually short and can be prepared only via agency of local team leaders and emails or phone calls, it can be hard to build close personal relationships with members of the distributed team. At the same time, research projects are often negotiated solely with the management, and the presence of researchers can be regarded as intrusive and interfering by remote team members (especially when employees fear consequences such as job losses). Hence, researchers studying Global Software Engineering are confronted with more problems than in qualitative research focusing on co-located settings, aspects like trust (and access negotiation) become even more problematic in such contexts.

Another situation that could influence the challenges we identified is the fact that research teams may be the research division of a large software company involved in GSE. Although our cases do not cover this situation, one of the authors had the chance to conduct GSE research in this context, and for this reason we believe that the challenges and recommendations do not change. In some situations the challenges can be easier to deal with, because you are dealing with colleagues from the same company. Access to company, understanding of organizational culture, usage of communication and collaboration tools and other aspects may be facilitated in this case.

We also found some challenges that may be specific for research in global contexts. First of all, GSE in terms of offshoring can be a highly political issue. After all, there is not only the need to access multiple sites, but several organizations, which may compete with each other in regard to wages, division of labour as well as interpretations and perspectives on shared projects (even if this is denied in interviews (Boden et al. 2007)). Hence, researchers risk getting involved in internal conflicts on site, as actors may try to take advantage of the research for their interests in the context of intra- or inter-organizational quarrels. Second, GSE is a field that is highly dependent on computer mediated communication, including a broad set of different tools, artefacts and media for work. While these rich sources of data also offer many opportunities, research undertaken in these complex settings can be very difficult in terms of getting access and analyzing the complex interrelations in order to get a detailed understanding of what is going on**.** Last but not least, we also found new challenges that were relevant for researching into Global Software Engineering research: the need of maintaining a global view from a local perspective, language as well as cultural issues.

5.1 Possible Solutions for the Exacerbated Challenges

Before detailing the particularities of distributed development work, it is important that we mention that many of the general recommendations for conducting qualitative studies are valid in this field too. First of all, any research (especially ethnographic studies) should avoid disrupting the software developers' organizational routines and should be clear about the expectations, requirements, and arrangements. In addition, as in any other empirical research, trust acquisition and access to companies can be facilitated when the possible gains that the organization could have from the research are pointed out. As many of the challenges have a highly social and situated component, it is also often more a matter of "soft skills" than a matter of rigor methodology rigor to successfully conduct qualitative research projects. In this regard, certain open-mindedness, a respectful demeanour, determination, and a 'can do' mentality are often helpful in overcoming most of the problems we will discuss below. Furthermore, tacit knowledge about how to deal with fieldwork in practice is an important component: while the basic principles and underlying theories can be taught in a graduate school, hands-on experience is necessary to become a skilled ethnographer. Just like teaching recipes does not make good cooks, learning ethnography usually happens in the form of master-apprentice relationships, and not by mere theoretical reflections about methodology (Gobo 2008).

Hence, the solutions we propose are based on our own experiences when conducting the studies presented in this paper, and need to be understood rather as "war stories" rather than as "best practice" recommendations for successful research projects in GSE.

5.1.1 Getting Access to Companies

With regard to the overall organization of research projects, the need to access multiple sites means that careful planning of the study is necessary and can help to deal with many of the inherent problems of conducting research in the field of GSE (Creswell 2003) as well as for avoiding misunderstandings and getting the approval of companies for data collection. Having prepared appropriate documents as hand-outs (project description, agreement of confidentiality, etc) can be of great benefit for the acquisition of industrial research sites.

In order to handle the problems related with getting access to companies, we found that getting in contact and collaborating with distributed researchers or professional associations that have access to distributed companies (De Souza et al. 2007) was extremely beneficial. Having established a good relationship with local companies also helped in coming into contact with practitioners (Prikladnicki et al. 2003). In De Souza and Redmiles's study (2011), he was part of the company during his data collection, with access to all communication and coordination tools, and that definitely facilitated his process of getting access (but not the one of gaining trust). If the brokering of established contacts is not sufficient there might be no solution than to visit conferences and trade fairs or search trade registers and yellow pages for possible partners.

In general, we found that trust was best fostered by initiating personal contacts to the practitioners and by participating in social rituals on site (like lunch or coffee breaks, waiting for the bus, etc.) (Boden et al. 2011). It can be a good strategy for researchers to introduce themselves to all team members and communicate goals and methods of their research openly. In order to maintain awareness it has also proven beneficial to have short presentations during team meetings reporting on the status of research on a regular basis. For example, Avram (2007a, b), Avram et al. (2009) participated in both weekly team meetings and management meetings over a period of 18 months and had periodically given short updates on her research in these meetings in order to maintain transparency and to encourage team members to volunteer passing information and documents to her.

5.1.2 Interviewing Team Members from Remote Locations

In order to facilitate the researcher's contact with members of the remote team, if possible, one should participate in the regular technology mediated meetings (via phone, Skype or other audio/video conferencing systems) where members of both the local and the remote team take part. A short introduction of the researcher and of the objectives of her\his study during such a meeting can create the necessary awareness and acceptance for making further contact easier. In the case of ethnographic studies, periodic short interventions of the researcher reporting on the status of the research in such joint meetings can help gaining the trust of the team members by providing some degree of transparency.

Furthermore, it can be very useful for the researcher to invest some time in building a relationship with her\his remote informants (Avram 2007a, b; Avram et al. 2009). Getting a better understanding of someone's role in the organization and a general awareness on the history of the collaboration between the sites can help the researcher to compare her\his observations against a broader context. For example, in the case of Avram et al., well-maintained personal pages, blogs, mailing lists, the team room and the organization chart provided valuable information for researchers (Avram 2007a, b; Avram et al. 2009). Once a good relationship is built with practitioners in remote locations, these practitioners can be used as informants whenever an incident occurs. Short, informal interviews via instant messaging can give the researcher the chance to compare her/his observations made from a local perspective against the perception of people at the remote site. It is also useful to maintain these relationships in case observations or interpretations need to be checked or re-interpreted in the light of new developments. Socializing with the informants might happen very naturally in collocated situations, while in distributed environments they require patience and additional effort in terms of communication.

5.1.3 Observing Team Members in GSE

In order to make the (usually short) visits to remote sites as efficient as possible, our experience suggest that scheduling trips to the distributed sites should be done later in the data collection phase when the researcher has more precise ideas of the data (s)he wants to collect. In addition, this strategy allows the researcher to get to know the offshore team-members by regularly participating in e-meetings (see previous section), thus building trust and making the entry into the distributed site easier. Sometimes, it is even possible to meet remote members of the team when they visit the local site. For example, in the case of Boden et al.'s study it was of great benefit that some of the Russian developers visited the German company during the project (Boden et al. 2007). Thus, the researcher had the possibility to meet and observe them for several days in Germany, slowly gaining their trust. This personal contact definitely made it easier to conduct participant observation at the remote site in Siberia later in the project. In another case, De Souza and Redmiles (2011) did not have another option than to schedule interviews with key participants when they travelled to the main site in the US. In this case, it should be noted that it was very important to schedule the interviews ahead of time and during lunch breaks because the remote developers were only going to spend a short period of time at the site and they had a very busy schedule.

5.1.4 Collecting Data from Documents

For accessing (restricted) systems, it can be very effective to ask an internal liaison to provide particular documents that are used or mentioned in e-meetings, conversations, or interviews: a specific e-mail that created controversy, a defect report at the origin of an argument etc. For example, in the case of Avram (2007b), an incident between a contract tester from France and the translation coordinator in Ireland was mentioned in one of the meetings. The researcher was able to obtain the chat transcript from the translation coordinator after explaining her interest in the artifact and its possible use in her study. This set of documents might need to be checked by the management before it is forwarded to the researcher. A solution adopted by de Souza was to ask team members to include him on the emails exchanged, which allowed him to collect important data (De Souza and Redmiles 2011). But, this was only possible because he was working for the same company of the developers and had access to their communication and collaboration tools. Further information can be obtained from documents publicly available: the company website, newspaper and magazine articles, blog posts speaking about the company.

5.2 Possible Solutions for the New Challenges

We have also identified new challenges that are not so easily dealt with: the necessity to obtain a global view, dealing with language issues, and culture. Unfortunately, the complexity of these challenges makes it hard to derive best practices, much less panacea. The adaptation of the solutions we presented may be problematic, as they are highly dependent on the specific parameters of the research project they need to be applied to. Large-scale projects with multiple researchers (perhaps from different locations) have a clear advantage here in comparison to small projects, but they also suffer from additional challenges that can arise from synchronizing the different research views (which, as we have pointed out above, can be subject to similar difficulties as the practitioners face in their daily work).

In the following sections, we will try to offer some insights from our experiences that might help other researchers in similar projects in tackling the issues that we found were important when conducting our work:

5.2.1 Obtaining a Global View

The best approach for obtaining a global view is probably to collect the data personally, i.e., researchers should try to organize trips to collect data face to face (observations and interviews, for example) as we did in some of our studies (Prikladnicki et al. 2003, 2007; Avram 2007a, b; Avram et al. 2009). Being at the remote site is important because it allows the researcher to be in contact with the rich work-environment of the distributed teams, and conduct face-to-face interviews asking informants to demonstrate what they are explaining during interviews. These insights help to understand the different perspectives involved in distributed projects, and can help to contextualize explanations and narratives from local project members.

Due to the importance of personal visits, it is especially important to plan projects in a way that would allow for travelling in terms of sufficient time and sufficient funding. Personal visits cannot be easily replaced by phone interviews or by interviewing local team members about the remote colleagues, so extended stays at the remote locations need to be taken into account early when writing a proposal or in the early stages of planning a project. Sometimes companies invite remote colleagues to the local site for meetings or project reviews; these are important opportunities for meeting remote team members in person, and can help in obtaining an alternative global view when travelling is not an option. If possible to arrange, cooperation with researchers active in the areas where the remote teams are located can also be an option, if there are working on similar research questions. In any case, researchers should stay aware about that their views can be biased, and account for this aspect in their analysis.

5.2.2 Dealing with Language Issues

When doing research in multi-national development projects, having on the team researchers that understand all the languages used for communication is a great advantage. Even when the shared project language is English, parts of the cooperation can be easily missed when the researcher does not understand the local language as we have pointed before. In order to motivate team members who are less fluent in the language of the project to participate in interviews, it can be advantageous to start with informal interviews that are not recorded, or communicate with them in a written form, because this is often perceived to be easier in comparison to face-to-face communication. It is also especially important to build trust with such team members, because they may be reluctant to admit their language skills limitations and hence be reserved towards the researchers.

In order to make observations of practitioners using a different language, it can be effective to include a person familiar with the language in the research team. This person does not necessarily have to be a researcher (Prikladnicki et al. 2008; Richardson et al. 2008) or a student (Prikladnicki et al. 2007; Boden et al. 2007; De Souza and Redmiles 2011). For example, researchers can try to hire a student from a local University at the research site to assist in the project for a short time, or involve a translator if the funding allow for such expenditures. If no translators can be involved, researchers will have to reflect carefully whether to encourage the practitioners to communicate in a language they understand (thus perhaps influencing the project communication), or account for several blind spots in their observation of project work, as in the examples outlined above.

5.2.3 Dealing with Cultural Issues

Dealing with cultural issues can be quite challenging due to the reasons we have discussed in the previous chapter. Cultural issues can influence projects in subtle manners that are hard to detect, and they can be hard to distinguish from other influencing factors, such as different social roles or asymmetric power relationships between the sites. In order to understand cultural issues, it is important to observe carefully and apply caution when attributing specific aspects to cultural differences. Asking questions that refer directly to cultural issues between the teams can be misleading, such aspects not being always obvious. Practitioners might be inclined to talk about national stereotypes and obvious differences such as religion or holidays. Often, such explanations say more about the interviewee than about the team interaction they are meant to describe. On the other hand, eliciting stories illustrating cultural issues can support a better understanding of the situation. In general, it is better to postpone such questions for later stages of the research, when trust has been established and when the researcher has a better idea about the common problems and issues of team cooperation.

If possible, joining social events such as joint lunches or nights can also offer a better grasp of cultural issues. On such occasions, professional protocols that often mediate coordination are put aside, and interaction of team members can be quite different as compared to at the workplace. For the same reason, conversations with team members about their personal aims, fears and expectations can provide a deeper understanding of the context and motivations of various team members. However, such conversations require a high level of trust and confidentiality between the researcher and the team members, so researchers need to be very careful when bringing up such aspects.

6 Threats to Credibility

In this section, we discuss the credibility of our experience report, building on (Onwuegbuzie and Leech 2007; Rigby and Storey 2011). As the paper is a summary of our experiences from a number of different research projects that have been conducted over the last decade in the context of Global Software Development, the study is building on a wide range of data sets. Appendix A provides details on the specific aims and methodologies of the case studies and ethnographies that formed the basis of our experiences.

In order to ensure that the issues we experienced in our studies were relevant in a broader set of research projects and had the potential to be transferred to other research projects, we developed them button-up by comparing different cases experienced by the authors, all having different backgrounds and experiences. We also contextualized our experiences referring to the related literature on Global Software Engineering and qualitative research methods in general.

Hence, although the experiences reported in this paper are based on our own specific research activities, and therefore, cannot be generalized to all qualitative GSE studies, we believe that the challenges and problems elaborated upon in this paper cover significant subset of themes relevant when conducting qualitative GSE research.

7 Summary and Conclusions

Global Software Engineering is changing the way people are acting as team members in software projects. With the increasing spread of GSE in the last years, many challenges are arising. Not surprisingly, these challenges are mirrored when conducting GSE research, and can be categorized as geographical, temporal, and cultural distances.

In our studies, we encountered several challenges that directly resulted from the spatial and geographic distribution of GSE projects. Many of the challenges we had to face in the context of GSE are common to other qualitative studies in the field of technology design, including the challenges of getting access, understanding complex organizational settings, building trust and dealing with the organization’s politics (Randall et al. 2007). However, as we conclude on the basis of our experiences, GSE as a field seems to exacerbate several of these problems. Hence, researchers have to take into account the specific nature of the distributed field and adjust their methodology accordingly, in order to better understand the practices they intend to support. Moreover, as we observed in Section 2, a substantial part of the research on GSE has to deal with understanding not only processes and plans, but also practices and people in order to develop better tools and approaches. It is very important to understand all the participants' points of view, instead of looking at the situation from one angle only. For this reason, interpretive methods such as qualitative methods are important for GSE as an enterprise. We need well conducted studies that adopt these multiple perspectives to advance the field; we believe that ignoring this need might be a challenge for further progress. In addition, it is important to note that the challenges and recommendations presented in this study are not unique to qualitative studies, and for this reason the paper could also be of interest for researchers conducting quantitative research.

Despite the profusion of GSE studies and a lot of accumulated experience from a research perspective, we were not able to find many documented experiences or possible solutions to overcome the existing challenges. This might be related to the fact that this type of research builds on practical experiences, usually difficult to explain and share. Hence, it is probably impossible to define a unique set of best methods for conducting research in GSE. However, it is possible to present "war stories", best practices and pitfalls from the field as examples illustrating the specific challenges of GSE research, in order to allow other researchers to learn from past experiences.

Our study represents a first attempt into this direction, synthesizing experiences from four different research groups located in three countries. We hope that other groups will follow in discussing methodological issues in order to improve GSE research through collaboration and experience sharing at the global level. In addition, we hope that such studies can help less experienced researchers and organizations dealing with critical challenges to gain insights in this relatively new research field.

Acknowledgments The studies reported in this paper were partially supported by: the Research Group on Distributed Software Development of the PDTI program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91); CAPES (project BEX 426006-6); CAPES (project BEX 1312/99-5); FAPERGS; CNPq (project 479206/2006-6, project 550130/2011-06, project 560037/2010-4, project 483125/2010-5); Science Foundation Ireland and Lero—The Irish Software Engineering Research Centre under PI grant 03/IN3/1408C; and the German Research Foundation (DFG) within the Project *ARTOS—Articulation work in Offshoring Projects of Small and Medium- sized Enterprises of the Software Branch.*

Appendix A—The Experiences of the Authors

This appendix provides details on the aims and methodologies of the various case studies and ethnographies which formed the basis of our experiences. Table 3 also shows the details of each of the experiences.

Table 3 Details of the experiences of the authors

Study 1: Avram (Avram 2007a, b; Avram et al. 2009)	Type: ethnographically informed study
Companies involved:	The Irish subsidiary of a Fortune 500 multinational company
Locations involved:	Ireland-Germany-US (East Coast)
Methods employed:	Ethnographic observation (70 full days over an 18-month period), 31 interviews(face-to-face and via instant messenger), online observation, document analysis;
Fieldwork in:	Ireland
Remote site(s) visited:	yes(Germany)
Study 2: Avram (Richardson et al. 2008; Boden et al. 2009, 2011)	Type: case study

Table 3 (continued)

Companies involved:	A SME specialised in software development for telecoms with headquarters in Ireland and a subsidiary in Romania
Locations involved:	Ireland, Romania
Methods employed:	5 interviews(face-to-face), observation, document analysis
Fieldwork in:	Ireland
Remote site(s) visited:	yes(Romania)
Study 3: Boden et al. (2007, 2008)	Type: ethnographically informed case study
Companies involved: partners in Rus	Two German SMEs engaging in offshore software development with partners in Russia
Locations involved:	Germany, Russia
Methods employed:	15 semi-structured interviews with managers and developers of different German SMEs, followed by semi-structured interviews (face-to-face and via Skype), participant observation, document analysis in the two companies
Fieldwork in:	Germany
Remote site(s) visited:	yes(Russia)
Study 4: Prikladnicki et al. (2007, 2008)	Type: exploratory case study
Companies involved:	Five multinational companies
Locations involved:	Brazil, Canada, US
Methods employed:	20 interviews (face-to-face and via Skype), accompanied by document reviews
Fieldwork in:	Brazil and Canada
Remote site(s) visited:	N/A
Study 5: Prikladnicki et al. (2003, 2006)	Type: case study
Companies involved:	Two Brazilian subsidiaries, each owned by a multinational organization
Locations involved:	Brazil, US, Europe
Methods employed: meetings minutes, and software development process descripti	22 individual interviews (face-to-face), document reviews, meetings minutes, and software development process descriptions
Fieldwork in:	Brazil
Remote site(s) visited:	no
Study 6: De Souza and Redmiles (2011)	Type: case study
Companies involved:	Large US software development company
Locations involved:	North Carolina, US; Massachusetts, US; Beijing, China; Shanghai, China; and Taipei, Taiwan
Methods employed:	17 interviews (face-to-face, phone, Skype), document analysis, including emails and instant messages exchanged among the software engineers, content of shared discussion databases
Fieldwork in:	US
Remote site(s) visited:	no

Study 1 In Avram's first study (2007a, b), Avram et al. (2009), the field site was an Irish subsidiary of a Fortune 500 multinational company involved in software development. Following a first contact with R&D managers, the request of following a software development team over an extended period of time was presented to a number of development managers, and one manager volunteered, with the agreement of his team. The study began in

January 2006 and the research team adopted an ethnographically informed approach. The first 3 months allowed the two researchers to familiarize themselves with the context and the work being done. In the following 15 months, they spent more than 70 full days in the field, observing the activity of the team in its own work environment, participating in meetings and group activities and occupying desks in the open plan in the team's area. 26 interviews were realised with members of the team based in Ireland, Germany and in the US, including the team lead, the two software architects, the lead of the QA team, software developers and quality engineers. Some of the team members were interviewed more than once. The research team was granted access to the company intranet, to the project's document repository and to the team's mailing list. They were also allowed to make use of the company's own instant messaging system, useful both as an awareness mechanism and as a communication channel with the members of the observed team. Observation and interaction continued online when the researchers were not present on site. Participation in teleconferences allowed them to observe directly the team members' interactions with people in various other locations (US, Germany, India). A good working relationship developed between the research team and the software development team and the researchers found many opportunities for conducting both formal interviews, and informal discussions on various topics—without disrupting the actual work. The researchers kept diaries, taking detailed notes on every day spent in the field. Remote collaborators of the observed team were also interviewed, either via instant messaging/phone or face-to-face. One of the researchers travelled to one of the company's sites in Germany in November 2006 and interviewed five people with different roles in the collaboration between the two sites.

The data collected from the field was periodically discussed and analyzed on a weekly basis by the extended research team, in order to identify topics, trends and problems and compare the findings to those from other similar sites where fellow research team members were observing similar processes and activities. The findings were shared with the members of the software development team, their managers and remote collaborators in conversations, specially designed workshops and in the form of draft reports and papers.

Study 2 In Avram's second study (Richardson et al. 2008; Boden et al. 2009, 2011), one of the cases presented is that of an Irish company with a development division in Romania. The method chosen for this study was that of a case study. One of the researchers, of Romanian origin, found the company website on the Internet. She contacted the Irish manager and organized an interview at the company headquarters in September 2007. After 2 months, she had the opportunity to travel to Romania, visit the Romanian branch and interview the Romanian manager (and co-owner) and 3 of the developers. At the time, the company was employing 3 project managers based in Ireland and Romania and 19 developers, all of them being located in Romania. Besides interviews, a limited amount of observation was undertaken in the two locations and a few documents were collected. Both the intermediary report and the finished paper were shared with the two managers and received positive feedback.

Study 3 In Boden's study (Boden et al. 2007, 2008), the researchers conducted an ethnographically informed case study in two German SMEs engaging in offshore software development with partners in Russia. The goal was to understand how software developers in distributed teams organize their development work in terms of Articulation Work, and how organizational learning can be organized in distributed settings.

The researchers started with an exploratory analysis of the literature on offshoring, covering discourses of various communities of practitioners and scholars. Based on these findings, relevant research questions were identified, focusing on informal and situated coordination practices of developers. On this basis, the researchers conducted 15 semi-structured interviews with managers and developers of different German SMEs. From this sample, two companies (named Alfa and Beta for the purpose of this study), which had offshored part of their software development to subsidiaries in Eastern Europe, were selected for closer investigation in the form of case studies.

For the case studies, the researchers conducted a triangulation of several ethnographic research methods, comprising of further semi-structured interviews, participant observation, as well as artifact analysis. For the participant observation, each of the German SMEs was visited for a period of ten to twenty working days. A third participant observation period was conducted at the Russian partner company of company Alpha—in Tomsk, Siberia. Beta's partner company could not be visited because of an abrupt end of the cooperation, but the researchers were able to cover the perspective of the Russian developers by conducting interviews via Skype with the Russian senior developer. During their time spent at the companies, the researchers had ample opportunities to observe local and distributed articulation processes, in the context of meetings, individual work situations and cooperative tasks. Several informal interviews were conducted and the researchers were allowed to analyze artifacts such as e-mails, chat protocols, internal work papers and whiteboard sketches.

The findings were documented by means of field notes and photos that were taken during the research. For validation, the findings were correlated with the literature focusing on articulation processes in software development. Furthermore, the identified concepts and topics were discussed with the participants of the study during a workshop.

Study 4 In Prikladnicki's first study (Prikladnicki et al. 2007, 2008) the authors have conducted an exploratory case study of distributed projects in five multinational companies. The data collection methods included interviews and document reviews. The documents reviewed were project plans, lessons learned, and documents describing the software development process. A total of 20 individual interviews (lasting 1 h each) included technical leaders, project managers, IT managers and directors. Interviews were conducted face-to-face -in Brazil and Canada, and over the telephone—with informants in the U.S. In Brazil, the access to the companies was facilitated by the researcher's previous contact. In Canada, the contact was facilitated by a professor living in that country, who had previous contact with the two companies. The professor was also part of the research team. In the U.S., the contact was made through the Brazilian subsidiary director, who has made the interview possible. The respondents were selected by convenience. Informants from three management levels: project management, information technology and portfolio management, and organization management were interviewed. Among them, there were six site directors, five information technology managers, seven project managers, and two technical leaders. The unit of analysis was the subsidiary. Findings were shared in the form of technical reports, papers, and also included in a book (the first book published in Portuguese on this theme).

Study 5 In Prikladnicki's second study (Prikladnicki et al. 2003, 2006), a case study was conducted in two software development subsidiaries, each one owned by a multinational organization with worldwide spread units. The first organization worked mainly in consulting, software development projects and training and had **external clients**. At the time when the research was conducted, it had nine software development subsidiaries located in Brazil. The organization also had offices located in Brazil and other countries in Latin America, as well as in the U.S. and Europe. Its headquarters were located in Brazil. The second organization supported and manufactured computers. It had three software development subsidiaries located in two continents that were responsible for **internal client demand** worldwide. The headquarters were located in the U.S. The data collection included primary and secondary sources. We conducted 22 individual interviews (11 in each organization), covering four projects, two in each organization. All interviews were conducted in Brazil, facilitated by previous connections between the companies and the research team. Secondary sources were also used: document reviews, meetings minutes, and software development process descriptions, together with public information available on the homepage of each organizations. For data analysis a content analysis protocol was defined and applied.

The respondents were selected according to the unit of analysis and the purpose of the study. Among the interviewees, there were project team members, development managers, quality assurance team members, software process improvement responsible and individuals representing the organizations' strategic level. Two questionnaires were developed, each considering a specific dimension to be explored: "organizational," containing information about the organization as a whole, and "project," with information on the four projects included in this study. One development manager was interviewed in each organization, whereas five interviews were conducted for each project. The convenience sample was not probabilistic, although the research team tried to get a good representation of all groups involved. Data collected was evaluated by practitioners in both companies—with positive feedback. Findings were used as an input for a training program on Global Software Engineering within one of the companies under study.

Study 6 In the de Souza's study (De Souza and Redmiles 2011), fieldwork was conducted in a large software development company that will be called LAR for the purpose of this article. LAR was one of the largest software development companies in the United States, with products ranging from operating systems to software development tools, including e-business and tailored applications. The project studied was responsible for developing a mobile application that had not been released yet. The project staff was divided into three major groups: user interface (UI) designers, software developers, and the quality assurance (QA) team. The staff was distributed over five different sites, spread in three different countries: North Carolina, US; Massachusetts, US; Beijing, China; Shanghai, China; and Taipei, Taiwan. To be more specific, user interface design and evaluation was performed by six professionals in North Carolina, and the implementation was performed in all other sites distributed as follows: nine developers in Massachusetts, five in Shanghai, five in Beijing, and four in Taipei. The quality assurance team was divided between U.S. and Chinese sites: three engineers were located in Massachusetts and six engineers in Beijing. The main coordination of the project and the project manager were located in Massachusetts, where all the data were collected.

Data was collected through document analysis and semi-structured interviews. Among other documentation, artifacts, emails and instant messages exchanged among the software engineers were collected. Access to shared discussion databases used by the software engineers was also granted for the research team. All of this information was used in addition to the notes generated by the interviews. We conducted 17 semi-structured interviews with members of all teams from the different sites: some interviews were conducted face to face, and others were conducted by telephone, with one interview conducted via instant messaging. The interview questions were designed to encourage the participants to talk about their everyday work, including work processes, problems, tools, communication, collaboration, and coordination efforts between their collocated and distributed colleagues. The interviews also aimed to explore the relationship between software dependencies and the coordination of software development projects, or, to be more specific, the potential usage of dependency information to facilitate collaborative software development.

Interviews lasted between 20 and 70 min. The interviews with some of the Chinese team members were conducted while they were visiting one of the US sites.

References

Aranda J, Venolia G (2009) The secret life of bugs: going past the errors and omissions in software repositories. Proceedings of the 2009 IEEE 31st International Conference on Software Engineering, IEEE Computer Society, pp 298–308

Argyris C, Putnam R, Smith DM (1985) Action science. San Francisco

Avram G (2007a) Of deadlocks and peopleware: colaborative work practices in global software development. In: ICGSE, Munich, Germany

Avram G (2007b) Knowledge work practices in global software development. The European Conference on Knowledge Management, Barcelona

Avram G, Wulf V (2011) Guest editorial: Studying work practices in Global Software Engineering, Information and Software Technology 53(9):949–954, ISSN 0950-5849, 10.1016/j.infsof.2011.01.010. (http://www.sciencedirect.com/science/article/pii/S0950584911000371)

Avram G, Bannon L, Bowers J, Sheehan A, Sullivan D (2009) Bridging, patching and keeping the work flowing: defect resolution in distributed software development. doi:10.1007/s10606-009-9099-6. Special Issue dedicated to 'Software Engineering as Cooperative Work' Guest Edited by Yvonne Dittrich, Dave W. Randall and Janice Singer, Journal of Computer Supported Cooperative Work, Volume 18, Numbers 5–6, pp 477–507

Biolchini J, Mian PG, Natali ACC, Travassos GH (2005) Systematic review in software engineering. Technical Report TR—ES 679 / 05, COPPE/UFRJ

Boden A, Nett B, Wulf V (2007) Coordination practices in distributed software development of small enterprises. In: ICGSE, Munich, Germany, pp 235–244

Boden A, Nett B, Wulf V (2008) Articulation work in small-scale offshore software development projects. Proceedings of the 2008 international workshop on Cooperative and human aspects of Software Engineering. ACM, Leipzig, Germany, pp 21–24

Boden A, Avram G, Bannon L, Wulf V (2009) Knowledge management in distributed software development teams—does culture matter? Proceedings of the 2009 International Conference on Global Software Engineering (ICGSE), Limerick, Ireland, pp 18–27

Boden A, Avram G, Bannon L, Wulf V (2011) Knowledge sharing practices and the impact of cultural factors: reflections on two case studies of offshoring in SME. J Softw Maint Evol Res Pract

Brown B, Lundin J, Rost M, Lymer G, Holmquist L (2007) Seeing ethnographically: teaching ethnography as part of CSCW. ECSCW 2007:411–430

Carmel E (1999) Global software teams—collaborating across borders and time-zones. Prentice Hall, Upper Saddle River

Carmel E (2006) Building your information systems from the other side of the world: How Infosys manages time zone differences. MISQ 5(1):43–53

Carver J, Seaman C, Jeffery R (2004) Using qualitative methods in software engineering. In: International Advanced School of Empirical Software Engineering (IASESE04), Los Angeles, CA, USA

Creswell JW (2003) Research design: qualitative, quantitative, and mixed methods approaches. SAGE Publications, USA

Cross R, Parker A (2004) The hidden power of social networks: understanding how work really gets done in organizations. Harvard Business School Press, 304 pp

Curtis B, Krasner H et al (1988) A field study of the software design process for large systems. Commun ACM 31(11):1268–1287

Damian D, Moitra D (2006) Guest editors' introduction: global software development: how far have we come? IEEE Softw 23(5):17–19

Dawson R, Bones P, Oates BJ, Brereton P, Azuma M, Jackson ML (2003) Empirical methodologies in software engineering. In: 11th Annual International Workshop on Software Technology and Engineering Practice, Washington, DC, pp 52–58

De Souza CRB, Redmiles DF (2011) The awareness network. To whom should I display my actions? And, whose actions should I monitor? IEEE T Software Eng 37(3):325–340

De Souza CRB, Hildenbrand T, Redmiles DF (2007) Towards visualization and analysis of traceability relationships in distributed and offshore software development projects. In: SEAFOOD, LNCS

de Souza CRB, Sharp H, Singer J, Cheng L-T, Venolia G (2009) Guest editors' introduction: cooperative and human aspects of software engineering. IEEE Software 26(6):17–19

Dittrich Y, John M, Singer J, Tessem B (2007) For the special issue on qualitative software engineering research. Inf Softw Technol 49(6):531–539

Dourish P (2006) Re-space-ing place: “place ” and “ space ” ten years on. In: Proceedings of the Conference on Computer Supported Cooperative Work

Dybå T, Prikladnicki R, Rönkkö K, Seaman CB, Sillito J (2011) Qualitative research in software engineering. Empir Softw Eng 16(4):425–429

Easterbrook SM, Singer J, Storey M, Damian D (2007) Selecting empirical methods for software engineering research. In: Shull F, Singer J (eds) Guide to advanced empirical software engineering. Springer

Espinosa JA, Nan N, Carmel E (2007) Do gradations of time zone separation make a difference in performance? A first laboratory study. In: ICGSE, Munich, Germany

Geertz C (1973) Thick description: towards an interpretive theory of culture. In: Geertz C (ed) The interpretation of cultures: selected essays. Basic Books, New York, pp 3–30

Gobo G (2008) Doing ethnography. Sage Publications, Los Angeles

Gonçalves MK, De Souza CRB, Gonzales VMG (2011) Collaboration, information seeking and communication: an observational study of software developers work practices. J Univers Comput Sci 17:1913–1930

Gupta A, Ferguson J (1997) Culture, power, place: explorations in critical anthropology. Duke University Press, Durham

Herbsleb JD (2005) Beyond computer science. 27th ICSE, 23–27

Herbsleb JD (2007) Global software engineering: the future of socio-technical coordination. 29th ICSE, 188–198

Herbsleb JD, Grinter RE (1999) Architectures, coordination, and distance: Conway’s Law and beyond. IEEE Software: 63–70

Herbsleb JD, Moitra D (2001) Guest editors’ introduction: global software development. IEEE Softw 18 (2):16–20

Herbsleb JD, Mockus A, Finholt T, Grinter RE (2001) An empirical study of global software development: distance and speed. In 23rd ICSE, pp 81–90

Hine C (2000) Virtual ethnography. Sage, London

Jacobson I, Booch G, Rumbaugh J (1999) The unified software development process. Addison Wesley Longman, Inc., Reading

Jarvenpaa SL (1998) Communication and trust in global virtual teams. J Comput-Mediat Commun 3(4). http://jcmc.huji.ac.il

King WR, Torkzadeh G (2008) Information systems offshoring: research status and issues. MIS Q 32(2):205–225

Klein HK, Myers MD (1999) A set of principles for conducting and evaluating interpretive field studies in information systems. MIS Q 23(1):67–93

Ko AJ, DeLine R, Venolia G (2007) Information needs in collocated software development teams. International Conference on Software Engineering (ICSE), Minneapolis, MN, May 20–26, 344–353

Kraut RE, Streeter LA (1995) Coordination in software development. Commun ACM 38(3):69–81

Kraut R, Egido C, Galegher J (1990) Patterns of contact and communication in scientific research collaborations. In: Galegher J, Egido C, Kraut R (eds) Intellectual teamwork: social and technological foundations of cooperative work. Lawrence Erlbaum, pp 149–172

Lethbridge T, Sim S, und Singer J (2005) Studying software engineers: data collection techniques for software field studies. Empir Softw Eng 10:311–341

Lutz B (2009) Linguistic challenges in global software development: lessons learned in an International SW Development Division. Global Software Engineering, International Conference on, Los Alamitos, CA, USA: IEEE Computer Society, pp 249–253

Marcus GE (1998) Ethnography through thick and thin. Princeton University Press, Princeton, New Jersey

Oates BJ (2006) Researching information systems and computing. Sage Publications, Thousand Oaks

Onwuegbuzie A, Leech N (2007) Validity and qualitative research: an oxymoron? Qual Quant 41(2):233–249

Paré G. Enhancing the rigor of qualitative research: application of a case methodology to build theories of IT implementation. The Qualitative Report, 7(4), retrieved from http://www.nova.edu/ssss/QR/QR7-4/pare.html, April of 2007

Parnas DL (2003) The limits of empirical studies of software engineering. In: Proceedings of the 2003 International Symposium on Empirical Software Engineering (September 30–October 01, 2003). International Symposium on Empirical Software Engineering. IEEE Computer Society, Washington, DC, 2

Patil S, Kobsa A, John A, Seligmann D (2011) Methodological reflections on a field study of a globally distributed software project. Inf Softw Technol 53(9):969–980. doi:10.1016/j.infsof.2011.01.013

Perry DE, Staudenmayer NA, Votta LG (1994) People, organizations, and process improvement. IEEE Softw 11(4):36–45

Plonka L, Sharp H, van der Linden J (2012) Disengagement in pair programming: does it matter? International Conference on Software Engineering, Zurich

Prikladnicki R, Audy JLN, Evaristo R (2003) Global software development in practice: lessons learned. J Softw Process Improv Pract 8(4):267–282

Prikladnicki R, Audy JLN, Evaristo R (2006) A reference model for GSE: findings from a case study. In: ICGSE 2006, Florianopolis, Brazil

Prikladnicki R, Audy JLN, Damian D, Oliveira TC (2007) Distributed software development: practices and challenges in different business strategies of offshoring and onshoring. In: ICGSE, Munich, Germany

Prikladnicki R, Evaristo R, Damian D, Audy JLN (2008) Conducting qualitative research in an international and distributed research team: challenges and lessons learned. In: HICSS 2008, Hawaii, USA

Randall D, Harper R, Rouncefield M (2007) Fieldwork for design: theory and practice. Springer

Richardson I, Avram G, Deshpande S, Casey V (2008) Having a foot on each shore—bridging global software development in the case of SMEs. In: Proceedings of the 3rd IEEE International Conference on Global Software Engineering, Bangalore, India, 19–21st August, 2008, IEEE Computer Society Washington, DC, USA, 13–22

Rigby PC, Storey MA (2011) Understanding broadcast based peer review on open source software projects. International Conference on Software Engineering, Hawaii

Rönkkö K (2000) Ethnography and distributed software development. In: ICSE Workshop Beg, Borrow or Steal: Using Multidisciplinary Approaches In Empirical Software Engineering Research, Limerick, Ireland

Seaman CB (1999) Qualitative methods in empirical studies of software engineering. IEEE Trans Softw Eng 25(4):557–572

Sengupta B, Chandra S, Sinha V (2006) A research agenda for distributed software development. In: 28th ICSE, pp 731–740, Shanghai, China

Sharp H, Robinson H (2004) An ethnographic study of XP practice. Empir Softw Eng 9(4):353–375

Spanoudakis G, Zisman A (2004) Software traceability: a roadmap. Handbook of Software Engineering and Knowledge Engineering. In: Chang SK (ed). World Scientific Publishing Co

Strauss A, Corbin J (1998) Basics of qualitative research: techniques and procedures for developing grounded theory, 2nd edn. Sage Publications, USA

Suchman L (1987) Plans and situated actions: the problem of human-machine communication. Cambridge University Press, New York

Taylor SJ, Bogdan R (1984) Introduction to qualitative research methods. Wiley, New York

Tosun A, Turhan B, Bener A (2008) Ensembe of software defect predictors: a case study. Proceedings of the 2nd Symposium on Empirical Software Engineering and Measurement

Wittel A (2000) Ethnography on the move: from field to net to internet. Forum Qualitative Sozialforschung, vol 1; http://www.qualitative-research.net/fqs-texte/1-00/1-00wittel-e.htm

Wohlin C, Höst M, Henningsson K (2003) Empirical research methods in software engineering. In: Conradi R, Wang AI (eds) ESERNET 2001-2003, LNCS 2765. Springer, pp 7–23

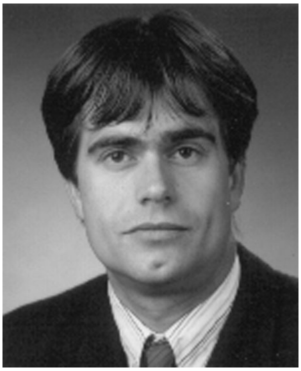
Wulf V, Rohde M, Pipek V, Stevens G (2011) Engaging with practices: design case studies as a research framework in CSCW. In: Proceedings of the 2011 Conference on Computer Supported Cooperative Work, pp 505–512

Rafael Prikladnicki is an associate professor in the Computer Science School and director of the Technology Management Agency (AGT) at Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) in Porto Alegre, Brazil. He also leads the Munddos research group. His areas of expertise are Distributed Software Development and Agile Methods for Software Development. Prikladnicki is mainly interested in how global software engineering impacts organizational decisions, the role of Brazil in the global IT industry, how global software engineering interplays with agile methods, and the usage of agile methods to build high performance software development teams. For more information, visit his website at www.inf.pucrs.br/~rafael.

Alexander Boden currently works as a post-doc researcher at the Institute for Information Systems and New Media of the University of Siegen. He holds a PhD in Information Systems from the University of Siegen, as well as a Master in Cultural Anthropology from the University of Bonn. In his dissertation, he investigated socio-cultural aspects of coordination and organizational learning in the context of software offshoring in small German enterprises. His research areas focus on Human Computer Interaction, Computer Supported Cooperative Work and Global Software Engineering.

Gabriela Avram is a Lecturer in Digital Media in the Department of Computer Science and Information Systems, and senior researcher at the Interaction Design Centre. She holds a PhD in Information Systems from the Academy of Economic Studies in Bucharest (Romania), and has undertaken postdoctoral studies in Germany, Luxembourg and Ireland with a focus on collaborative work in Software Engineering teams. For the duration of the socGSD research project at the University of Limerick (2005-2008), she studied collaborative work practices in cross-cultural environments, working with IBM, HP and Intel, as well as with a number of SMEs.

Cleidson R. B. de Souza received his B.Sc. in Computer Science from the Federal University of Pará, Brazil in 1996 and his M.Sc. in Computer Science from State University of Campinas, Brazil in 1998. He received his Ph.D. in Information and Computer Science in 2005 from University of California, Irvine. He is an Associate Researcher at the Vale Institute of Technology and an Associate Professor at the Federal University of Pará, Brazil. From 2010 to 2012, he worked as a Research Scientist at IBM Research Brazil. He has a background in software engineering and computer-supported cooperative work, and his research interests are in the intersection of these areas.



Volker Wulf is a professor at the University of Siegen and at Fraunhofer FIT, St. Augustin, Germany. His research interests lie primarily in the area of Human Computer Interaction and Computer Supported Cooperative Work, recently also in Global Software Engineering. For more information, visit his website at http://www.wiwi.uni-siegen.de/wirtschaftsinformatik/mitarbeiter/wulf/index.html.en.