

Persistent Discontinuities in Global Software Development Teams: Adaption through Closely Coupled Work Practices

DISSERTATION

From the Faculty of Technologies in Practice at the IT
University of Copenhagen

DOCTOR OF PHILOSOPHY

In Information and Computer Science

By

Rasmus Eskild Jensen

Dissertation Committee:
Associate Professor Irina Shklovski
Professor Volker Wulf
Professor Anne-Marie Søderberg

2014

To Annalin

ACKNOWLEDGEMENTS

During my three years as a PHD student, I have met many helpful people who have guided me and provided thoughtful insights on how to proceed with my research.

Firstly, I would like to thank Professor Bonnie Nardi for welcoming me at UC-Irvine. I have really enjoyed working with Bonnie since we did not only share a professional context but also both had a passion for digital games. It was very inspiring to work with Bonnie and it definitely helped me clarify the purpose of my research. Moreover, being part of her research group was very encouraging and I learned a lot from the presentations and discussions in the group.

Secondly, I want to acknowledge a group of people who have helped me one way or the other during my PHD studies. Dave Randall has been very supportive and always willing to discuss my work when I had questions. Debra Howcroft and Nina Boulus greatly improved my writing and reviewing skills when submitting papers. Paul Dourish and Ojelanki Ngewyama provided critical feedback and comments to push the research forward during the essential stages of my PHD. I also want to thank Dan Cosley for sharing his experiences of the networking aspect at conferences. I would also like to thank the entire NexGSD team for being great colleagues and for all the interesting workshops and seminars held over the years. Many more researchers from the CSCW community have helped me directly or indirectly just by being open minded and friendly towards me when I first joined the community. I can honestly say that the CSCW community has a very unique and friendly atmosphere that I came to treasure a lot over the years.

I guess no one gets through a PHD without having someone to share the successes as well as the frustrations with and luckily I have had some very good PHD colleagues in my department. Stina, Andreas, Morten, James, Lea, Louise, Olivier, Anne-Katrine, Nikki and Neha have all been part of making the last three years much more enjoyable.

Last, but definitely not least, I would like to thank my supervisor, Pernille Bjørn, for being supportive and constructive throughout all the different stages of my PHD. I have felt lost many times during my three years of becoming a researcher but every time Pernille guided me to new possibilities in my research and was thus able to push my work forward.

On a final note I want to thank my close friends and family for bearing with me during three years of what can be characterized as temporary insanity and for laughing with me when things looked disastrous.

PARTICIPATION IN THE CSCW COMMUNITY

This is a dissertation of publications and the main part is a selection of articles, which will be presented in the next chapter. However, I have participated in the CSCW community with work that will not be presented in the main part but is included in this chapter. I have participated in a range of CSCW conferences, more notably the CSCW, GROUP, COOP and ECSCW conferences. Moreover, I have participated in conferences in the periphery of the CSCW community namely the ICIC (now CABS), ECIS and IRIS. My participation in the community includes paper presentations, workshop participation, posters, doctoral colloquiums and student volunteering. Below is a full list of my contributions to the CSCW community.

Jensen & Bjørn, P. (2011): Global Software Development: The Complexities in Communicating about the Requirement Specification across Culture and Geography, *Presented at the IRIS Conference*, August 2011, Turku, Finland.

Jensen, R. E., & Bjørn, P. (2012). Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds. *Presented at the 10th International Conference on the Design of Cooperative Systems (COOP)*, May 2012, Marseilles, France.

Jensen, R. E. (2013). Commitment manifested in activity: A non-instrumental approach to commitment in virtual teams. *Position paper for Workshop on Trust in Virtual Teams: Theory and Tools at the Conference for Computer-Supported Cooperative Work (CSCW)*, February 2013, San Antonio, Texas. USA.

Jensen, R. E. (2014): "Transfer of Responsibility and Building Trust in a Global Software Development Project" *Position paper for Workshop on Global Software Development in a CSCW Perspective, Tools at the Conference for Computer-Supported Cooperative Work (CSCW)* February 2014, Baltimore, Maryland, USA.

Christensen L. R., Jensen R. E., & Bjørn, P. (2014). Creating Relation Work: Characteristics for Local and Global Collaboration *Presented at the 12th International Conference on the Design of Cooperative Systems (COOP)*, May 2014, Nice, France.

Jensen, R. E., & Nardi, B. (2014). The Rhetoric of Culture as an Act of Closure in a Cross-National Software Development Department. *Presented at the European Conference on Information Systems (ECIS)*, June 2014, Tel Aviv. Israel.

Jensen, R. E., & Bjørn, P. (2014). Demonstrating Commitment in Practice: Tracing & Gaining Attention. *Submitted to the Journal of New Technology, Work and Employment*.

Jensen, R. E. (2014). Why Closely Coupled Work Matters in Global Software Development. *Presented at the International ACM Conference on Supporting Group Work (GROUP)*, November 2014, Sanibel Island, Florida, USA.

COLLECTION OF PUBLICATIONS

This doctoral dissertation is composed of a collection of publications with an introduction, so that the main part of this dissertation is presented as a selection of four peer-reviewed papers, which are all included at the end. The introduction includes an extended methods chapter and a theoretical framework to place the results of the articles in the wider context of the current state of the research. The introduction is not expected to provide new results, but it will provide a synthesis of new conclusions by combining the results from the papers. The four papers included in the dissertation are shown below in chronological order.

Paper No. 1: Jensen, R. E., & Bjørn, P. (2012). Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds. *Presented at the 10th International Conference on the Design of Cooperative Systems (COOP)*, May 2012, Marseilles, France.

Paper No. 2: Jensen, R. E., & Nardi, B. (2014). The Rhetoric of Culture as an Act of Closure in a Cross-National Software Development Department. *Presented at the European Conference on Information Systems (ECIS)*, June 2014, Tel Aviv, Israel.

Paper No. 3: Jensen, R. E., & Bjørn, P. (2014). Demonstrating Commitment in Practice: Tracing & Gaining Attention. *Submitted to the Journal of New Technology, Work and Employment*.

Paper No. 4: Jensen, R. E. (2014). Why Closely Coupled Work Matters in Global Software Development. *Presented at the International ACM Conference on Supporting Group Work (GROUP)*, November 2014, Sanibel Island, Florida, USA.

Papers No. 1, 2 and 4 have already been peer-reviewed and accepted for publication. Paper No. 3 is under review for the *Journal of New Technology, Work and Employment*.

ABSTRACT

Collaboration in global software development projects can be very challenging due to discontinuities such as geographical distance, cultural diversity and changing team composition making the global collaboration work, despite these discontinuities, is important in order to achieve the proposed benefits of lower cost, scalability and possible cross-cultural learning in the teams effectively. The literature proposes that discontinuities eventually can turn into a constant state or *continuities* when routine and practices have found a state that can handle the discontinuity. However, the findings from this dissertation suggest that global software development teams experience frequent changes as normal natural trouble in the collaborative work.

Investigating the data material from an ethnographic study of a Danish-Philippine software development project, the data showed that frequent changes in the team composition create persistent discontinuities making it challenging to establish a shared context and mutual knowledge. Thus discontinuities did not reliably transition into continuities over time, because the establishment of shared norms and expectations was disrupted when IT developers repetitively left the project. Despite these persistent changes the project members managed to create coordinative practices allowing them to experience the best collaboration to date. Using this as a starting point, it is clear that researchers still know little about how practitioners adjust and adapt to persistent discontinuities in globally distributed teams or how practitioners coordinate the work to bridge persistent discontinuities? This dissertation contributes to these questions by asking the following research question: *How do IT developers coordinate the work to facilitate frequent changes in global software development projects?*

The data material revealed a gradual shift towards more closely coupled work practices over the course of three years as practitioners adjusted to make the collaboration work. Closely coupled work became a method for the practitioners to coordinate the work and reduce the complexity of discontinuities. Mutually shared financial responsibility between the Danish and the Philippine offices was the first step towards more interdependence between the Danish and Philippine project members. Secondly, the organizational structures must be geared to facilitate global collaboration with flexible travel policies to alleviate the risk of emergent negative sub-group dynamics. Lastly, the coordination of tasks should emphasize mutual dependencies across locations to establish an incentive for more interaction between the project members. Mutual dependencies enabled closely coupled work practices, which established connections across both locations and organizational hierarchy. These connections made the team more resistant to frequent changes in the team composition and made it easier to trace collective attention in the everyday work.

In conclusion, the findings showed that changes in the organizational structures alone such as the contractual setup and the travel policies were not enough for making the collaboration work, but combined with closely coupled work practices allowed the practitioners quickly to adjust and react to frequent changes in the project. Sharing dependencies facilitated an easier learning process for new project members and was thus essential for completing the task.

Table of Contents

1. INTRODUCTION	8
1.1. RESEARCH QUESTION	9
1.2. THE NEXGSD RESEARCH PROJECT	11
1.3. READING GUIDE	12
2. METHOD	12
2.1. THE EMPIRICAL CASE	14
2.1.1. GLOBALSOFT	14
2.1.1. THE DAFIC PROJECT	15
2.1.2. THE DANISH OFFICE	15
2.1.3. THE PHILIPPINE OFFICE	15
2.2. THE DATA SOURCES	16
2.3. THE ANALYTICAL APPROACH	18
2.4. THE ROLE AS A RESEARCHER IN AN ETHNOGRAPHIC WORKPLACE STUDY	21
2.5. REFLECTIONS ON MY ROLE AS AN OBSERVER IN THE DAFIC PROJECT	21
3. INVESTIGATING GSD FROM A CSCW PERSPECTIVE	24
3.1. UNDERSTANDING GLOBALLY DISTRIBUTED WORK	25
3.2. THE EXTRA EFFORT REQUIRED TO SOLVE THE TASK	26
3.3. MAKING WORK PRACTICES VISIBLE	27
3.4. COORDINATION PRACTICES	29
3.5. ESTABLISHING A MUTUAL UNDERSTANDING ACROSS DIFFERENT SOCIAL WORLDS	32
4. SUMMARY OF THE PAPERS	36
4.1. PAPER NO. 1: DIVERGENCE AND CONVERGENCE IN GLOBAL SOFTWARE DEVELOPMENT: CULTURAL COMPLEXITIES AS SOCIAL WORLDS	36
4.2. PAPER NO. 2: THE RHETORIC OF CULTURE AS AN ACT OF CLOSURE IN A CROSS-NATIONAL SOFTWARE DEVELOPMENT DEPARTMENT	37
4.3. PAPER NO. 3: DEMONSTRATING COMMITMENT IN PRACTICE: TRACING & GAINING ATTENTION	38
4.4. PAPER NO. 4: WHY CLOSELY COUPLED WORK MATTERS IN GLOBAL SOFTWARE DEVELOPMENT	38
5. DISCUSSION	40
5.1. PERSISTENT DISCONTINUITIES IN GLOBAL SOFTWARE DEVELOPMENT TEAMS	41
5.1. INCREASING DEPENDENCIES WITH CLOSELY COUPLED WORK	42
5.2. CHARACTERISTICS OF CLOSELY COUPLED WORK	43
5.3. ADJUSTING TO FREQUENT CHANGES IN THE COLLABORATIVE WORK	44
6. CONCLUSION	46
7. REFERENCES	48
8. MAIN CONTRIBUTION	55

1. INTRODUCTION

The benefits of global collaboration for software development teams seem to be very advantageous at first sight. The advantages include lower cost, increased scalability and competitiveness on the global market as well as providing access to new markets, a global workforce and enabling innovative synergies (Conchúir et al. 2009). However, the disadvantages such as working across geographical distances, time zones and culture may offset some of these advantages. The overlay in communication needed often increases costs and differences in cultural backgrounds can cause misunderstandings (Holmström, Conchúir, et al. 2006a). These difficulties may eventually lead to lower quality of the end product (Noll et al. 2010). Thus globally distributed collaboration increases the complexity of the everyday work and requires transitions in roles and competences to make the collaboration work (Matthiessen et al. 2014).

Time and time again, practice has shown that working across geographical distances remains a challenge for the collaboration (Judy S Olson & Gary Olson 2013). Companies apply different strategies to overcome the challenges with various degrees of success depending on the type of work and the tasks. Some companies are committed to strict process control such as the capability maturity model (Ebert & De Neve 2001) while others have a more flexible approach (Holmström, Fitzgerald, et al. 2006b). While global collaboration between software developers is not entirely new and can be dated back to the beginning of 1980 (Lonsdale & Cox 2000), many challenges remain despite new strategies and technological improvements. The complexity of developing software in constantly changing work environments creates emergent challenges that the practitioners have to deal with to make the collaboration work. Thus it remains a key interest for both researchers and industry to come to understand not only the core challenges global software development (GSD) but also to point to undiscovered opportunities of working across temporal, geographical, and cultural differences (Noll et al. 2010; Aspray 2006; Šmite et al. 2010).

Global software development practices represent an interesting field of study for researchers trying to investigate the work practices of geographically distributed teams. Researchers have pointed to the extra communication required when teams span *discontinuities* such as geographical distances, time differences and cultural diversities. Discontinuities are defined as ‘gaps or lack of coherence in aspects of work such as work settings, tasks or relations with other workers and managers (Watson-Manheim et al. 2002, p.193). Geographical distances separate people and impede day-to-day interactions between colleagues since face-to-face meetings are less likely to take place due to the excessive costs. This is the reason why communications are technology mediated in globally distributed teams (Herbsleb et al. 2001). Working across different time zones may disrupt work practices and requires more coordination when teams work asynchronously. Some teams attempt to coordinate work around a 24-hour cycle (follow-the-sun) or to reschedule working hours so that teams can work partly synchronously (Treinen & Miller-Frost 2006). Coordinating the work around 24-hour schedules is typically most efficient when the task is standardized and has few dependencies, such as IT support call centers (Carmel and Tjia 2005). Cultural diversity is another kind of discontinuity that global

software teams may encounter in their daily work practices. Studies of cultures in cross-national teams have focused primarily on how to mitigate the problems of cultural differences (Kayan et al. 2006, Diamant et al. 2008, Deshpande et al. 2010). The findings from these studies suggest that cultural differences can result in misunderstandings and breakdowns due to differences in language, norms and the fact that practitioners may originate from radically different social backgrounds. To mitigate the impact of different cultures, global software development research has pointed to cultural education or using people as boundary spanners to bridge the cultural gap (Krishna et al. 2004).

Thus discontinuities create a lack of understanding in a collaborative work setting, which may disrupt the work practices in several ways and practitioners in global software teams have to find ways to close the gap. Watson-Manheim (2002) propose that when ‘those participating in virtual work arrangements become familiar with each other, develop shared norms and expectations, and so forth, what was perceived as a discontinuity at one point in time, may later be perceived as a continuity’ (p. 202). Continuities represent a stable state of work where the participants in distributed teams have adjusted to the discontinuity so that it does no longer pose a challenge for the everyday work practice. Continuities emerge when the practitioners find ways to bridge the differences that emerge with discontinuities and create a shared coherence. Interestingly, these findings propose that discontinuities such as geographical distance can be ‘routinized’ over time in geographically distributed teams. The gaps and lack of coherence caused by discontinuities may be transformed by new work practices that enable continuities to emerge as time passes. Consequently, discontinuities can be perceived as continuities when the practitioners have had time to establish shared norms. However, the time required for establishing shared norms may rarely be available in GSD teams because of constant changes in team compositions. This dissertation questions the notion of continuities and proposes that discontinuities are persistent in global software teams because frequent changes in the team composition make it difficult to establish shared norms. However, this is not to say that the effort to *adjust* will not prevail but it requires shared financial responsibilities, enabling organizational structures and mutual dependencies in the teams.

1.1. Research Question

Although studies of global software development are increasing there is still a need for more ethnographically informed studies performed in industrial settings (Šmite et al. 2010). We still know little of what practices are effective and in what settings and Šmite points to the need for looking beyond the individual solutions of tools, practices and methods when researching GSD (Herbsleb 2007) and there is still much to gain from more in-depth studies of the work practices in global software teams. In recent years researchers have engaged in workplace studies of global software teams both cross-sectional and longitudinal and provided rich insights into this specific area of work (Prikladnicki et al. 2003; Avram et al. 2009; Boden et al. 2007; Matthiessen et al. 2014). The studies on global software development teams have provided us with much needed information about various aspects of the work practices in global software development teams. However, we still know little about the conditions that enable the practitioners to mitigate the lack of coherence in globally distributed teams. This question still remains largely unanswered throughout the literature on global software development and to investigate it properly, researchers need to move beyond

‘the use of students in research’ and instead to study global work practices in actual work settings over prolonged periods of time (Watson-Manheim et al. 2002, p.202). Cross-sectional case studies will not be able to report on changes in work practices over time and investigate how or if discontinuities become routinized over time. Moreover, experimental settings using students also lack the element of process as well as being able to replicate the intricate complexities of actual work practices. In organizationally collocated settings, work practices are dynamic and changing as Schmidt and Bannon pointed out in 1992:

The conception of organizational work, where people perform a number of tasks according to a set of well-specified “procedures”, has been proved to be highly idealized and grossly inadequate for analyzing and modelling the articulation of real world cooperative work arrangements. Due to the dynamic and contradictory demands posed on a social system of work by the environment, task allocation and articulation are renegotiated more or less continuously (Schmidt & Bannon 1992, p.16).

Adjusting to a dynamic work environment in distributed collaborative settings involves the renegotiation of tasks and procedures as discontinuities emerge or change into accepted continuities. The adjustment to new practices does not seem to happen automatically rather it has been suggested that change is triggered by emergent events that require immediate action from the practitioners (Tyre & Orlikowski 1994). Such discrepant events could for instance be a major breakdown in the collaboration or critical delays in the project planning (Majchrzak et al. 2000). A common discrepant event in global software teams occurs when people leave the team causing a loss of knowledge and a need to repair social connections (Boden et al. 2009). Similarly, changes in locations or organizational structures are also discontinuities that the IT developers have to face. In these instances quick technological ‘fixes’ and ‘best practice’ may not address the core of the issues in global software development. Instead recent research suggests that the efforts of making global software work involve a learning process over time as participants engage in the collaboration (Boden et al. 2012). To move this further, there is an urgent need to identify and understand the practices involved in adjusting to existing discontinuities, such as frequent changes in the team configuration. This dissertation contributes by investigating a single case study of a global software development project lasting three years asking the following research question:

How do IT developers coordinate the work to facilitate frequent changes in global software development projects?

The research question is a result of a continuous analytical process. The initial research question was open-ended and sought to investigate challenges and opportunities in global software development teams. During the analysis of the data, several subsidiary research questions emerged and were addressed individually in the papers. These questions spanned different aspects of challenges and opportunities in the global work, such as understanding the differences in social worlds (Paper No. 1) or the importance of closely coupled work (Paper No. 4). Thus the process illustrates an overall learning process leading to the research question that is the basis for this dissertation. Investigating how IT developers adjust to frequent changes became a pivotal topic in this dissertation and provides an overall framework for the findings in

the papers. Frequent changes are inevitable in dynamic global software companies and this dissertation identifies the enabling conditions that allow practitioners to adjust to these changes through closely coupled work. Moreover, this dissertation investigates how closely coupled work facilitates the collaboration in a setting with persistent discontinuities. Answering these questions will provide a deeper understanding of the coordination of work practices occurring in a global software development team over time when practitioners adjust to constant discontinuities.

1.2. The NexGSD Research Project

The work presented here is part of a larger research project called ‘NexGSD - Next Generation Technology for Global Software Development’ and some of the data material is collected in collaboration with colleagues. The NexGSD project seeks to develop next generation technologies – infrastructure, tools, and methods – that will bridge geographical, temporal, and cultural differences in Global Software Development. The overall aim of this project is to understand and improve GSD practices and technologies on a practical/industrial level as well as on a scientific level. The NexGSD research project will aim at promoting three main agendas on both a practical and an industrial level: 1) To *understand* challenges and opportunities of temporal, geographical, and cultural differences and develop new ways of conceptualizing and coping with cultural differences when managing complex GSD projects; 2) To *design* and empirically evaluate next generation GSD technologies that view GSD as collaboration rather than as outsourcing, and help software developers and end-users maintain a global awareness and a commitment to bridge distances in software development; 3) To *develop* and empirically evaluate a framework of GSD processes, practices, norms, and guidelines appropriate for GSD projects.

To achieve these research agendas, 15 researchers with interdisciplinary backgrounds have collaborated with seven industrial partners of various sizes. After investigating several different industrial settings of global software development using ethnographically informed methods, the researchers were able to develop informed insights from multiple sources and to compare case studies. Ethnography gives the researcher detailed descriptions of routines and daily work procedures and it is also an ‘opportunity to open up the overall problem-solution frame of reference in the context of some proposed solutions to specific identified problems’ (Anderson 1994). Ethnographically informed studies of industrial settings develop both a deeper understanding of the research area and potentially guide the design and development of tools and frameworks for GSD projects. By looking at practices and thus investigating the taken-for-granted assumptions, this dissertation contributes with deep insights that may guide the design of new tools and processes in global software development. However, the focus of this dissertation is *not* the design of new tools. Rather this work contributes primarily to the first agenda in the research project, namely to *understand* the challenges and opportunities in a GSD team spanning temporal, geographical and cultural distances. Thus the focus of this dissertation is not to determine parameters of success or failure from a normative perspective. Instead this dissertation investigates the everyday practices of software developers from a CSCW perspective in order to get a better understanding of the eminent challenges of teams engaged in global software development work practices. The starting point for this dissertation was to investigate work practices from a CSCW

perspective related to problems embedded in the research field of global software development.

This dissertation contributes to both of the overall research agendas in the NexGSD project. Firstly, the CSCW perspective used in this dissertation seeks to investigate the basic nature of collaboration and the goal has been to understand the challenges and opportunities of developing software in a geographically distributed setting, thus distinctly providing a contribution to the first of the stated research goals by identifying and conceptualizing the main challenges, including cultural diversities in global collaborative projects (see also Paper No. 2). Secondly, the CSCW perspective focuses on the design and development of new technology and work processes that can support the collaborative practices. While I have not provided distinct design solutions, my work has contributed through on-going presentations and discussions with other NexGSD project members. In fact, I have been closely involved in the development of ideas for supporting relationship work in meetings using tablet based technology (see also Christensen L. R., Jensen R. E., & Bjørn, P. 2014) and I have also provided insights and empirical examples for the development of digital scrum boards. In conclusion, this dissertation makes a distinct contribution to the NexGSD project not only in terms of identifying key challenges and opportunities for global collaborative practices in software development projects but also in terms of providing insights and empirical grounding for design solutions.

1.3. Reading guide

This dissertation is structured as two parts. The first part is the introduction and the second part is the main part, which is a collection of four peer-reviewed papers. The introduction is structured as follows: The following chapter introduces the empirical case, the methods and the data collection as well as a reflection on my role as a researcher in a workplace study. Chapter 7 provides the theoretical framework for the four papers. Chapter 8 is an introduction to the four papers and their contribution. Chapter 9 discusses the research question by linking it to the findings across the four papers and finally Chapter 10 is the conclusion.

2. METHOD

The methodological approach for this dissertation was an ethnographic workplace study (Luff et al. 2000, Randall et al. 2007) of globally distributed practitioners involved in software development. Computer-Supported Cooperative Work (CSCW) research has a long tradition of workplace studies (Heath & Luff 1992; Bentley et al. 1992; Bjørn & Christensen 2011). Ethnographic workplace studies rely on data collection in real world settings (Blomberg & Karasti 2013). Ethnographic workplace studies enable researchers to identify ‘invisible’ work practices and efforts that are not accounted for in verbal accounts’ (Luff et al. 2000; Blomberg & Karasti 2013). The field site of this study includes the collaborative work practices between practitioners located at two geographically different sites and thus the boundaries of the field site can be defined by the collaborative work practices between the Danish and Philippine employees.

Before going into the field, my initial research question was to understand the challenges and opportunities of global software development. In other words I wanted to understand *why* the practitioners found it challenging and *how* to overcome these challenges by conceptualizing opportunities. Such questions require a deep understanding of the work practices and the motives behind particular actions and such knowledge could best be obtained by applying an ethnographic approach to the subject of inquiry since ethnographic methods strive at understanding the meaning of certain activities. Thus the work of this dissertation is based on an ethnographic fieldwork study of two geographically separated locations for an extended period of time. This kind of work comprises some clarification in terms of what “ethnography” means and how the study of ethnography is used in geographically separated places. In this section I will address them each in turn. Ethnography is a term that is being used very broadly within the social sciences, such as anthropology and sociology, and it is also frequently used in CSCW research. Ethnography has become an integral part of research in CSCW and thus little effort is spent on defining the underlying assumptions behind ethnographic research. However, these assumptions are important and will be clarified here. Ethnography has been defined as “the study of people in naturally occurring settings or fields” by methods of data collection, which capture their social meanings and ordinary activities, involving the researcher participating directly in the setting, if not also the activities, in order to collect data in a systematic manner but without meaning being imposed on them externally” (Brewer 2000, p.6).

The basic assumption of ethnography is that is a ‘*method for understanding what activities mean to the people who do them*’ (Harper 2000, p.245). So what does it mean when ethnographers want to understand the meaning behind activities? Firstly, it is necessary to take a step back and identify at the specific ontological and epistemological perspectives of how ethnographers investigate the social world. The ontological assumptions behind ethnographic work, question the ontological assumptions of positivism in the social world, namely that social life and society itself are centred around objective structures independent of the consciousness of the individuals (Brewer 2000). In the positivistic tradition it is assumed that social life can be identified only through what is externally observable through the senses and from there it is possible to develop law-like statements about the social world. Opposite to these views, ethnographers propose that social phenomena are in fact not “objective” and cannot be rendered untouched by people’s interpretative capacities. This is also what is referred to as the hermeneutical paradigm that criticises the notion of an “objective” observer untouched by the subject in question. Object and subject cannot be separated especially when studying social phenomena where people are meaning-endowing and capable of constructing their own social setting (Brewer 2000). Social phenomena are wholly or partially constructed and reconstructed on the basis of interpretative processes performed by people. Thus society cannot be seen as a static entity but is interdependent of context and existing structure.

This ontological perspective of the social world naturally leads to other epistemological assumptions for the ethnographer. The focus of the ethnographer is not to test hypotheses and establish cause and effect relations of social phenomena, but rather experience and observe what is naturally happening in a setting. Thus the ethnographers apply a wide range of tools to comprehend what is going on in the society or in a certain setting. These tools or methods include most prominently participant observation and semi-structured interviews but a range of other methods is

also commonly used for instance workshops, cultural probes and presentations. It is most important for the ethnographer to remain flexible in the field and avoid a strict or rigid approach to the methods applied since this might exclude the researcher from revealing interesting findings (Feyerabend 1975). Thus I collected data from many different sources, both traditionally through interviews and participant observation, but also by recording screen activities and presenting initial results to the practitioners. I believe that this approach gave me a deep insight into the work practices and also helped me to establish a bond of trust among the participants.

2.1. The Empirical Case

Investigating how distributed team members coordinate their work practices to adjust to frequent changes in a global software development company, requires extensive access to an empirical setting for a longer period of time. Moreover, the empirical setting should also allow the researcher to follow all the details of work practices in global software development teams. Now as it turned out the empirical work was conducted in an organization referred to as GlobalSoft (a pseudonym). GlobalSoft proved to be the perfect match for this investigation since they had just begun a software development project spanning both Danish and Philippine software developers. The main empirical work conducted in GlobalSoft consists of an in-depth investigation of one particular project in GlobalSoft: DAFIC (DANish-FILipino Collaboration). The DAFIC project was ideal from a research perspective for several reasons. Firstly, the project engaged in close collaboration across discontinuities such as geographical distance, time and cultural differences. Secondly, the complexity of the tasks could not be handled by formal processes alone and thus required a lot of articulation work and coordination. The DAFIC project developed solutions for governmental practices and it was critical for the practitioners both in Denmark and the Philippines to understand the highly customized nature of the solutions. Moreover, working with the Danish municipalities on customized solutions had profound consequences for the level of documentation, the requirement specification and thus the distributed collaboration with the Philippines. Lastly, the project had just begun when the research was initiated and the project was scheduled to end after one year thus giving a reasonable time span for investigating the possible adjustments to existing discontinuities. Given the fact that the DAFIC project got extended for an additional two years, provided an even better opportunity to investigate how distributed team members coordinated their work adjustments to the frequent changes over time in a global software development project.

2.1.1. GlobalSoft

GlobalSoft is a software development company incorporated in Denmark in 1994. The company had total revenue of 241 million EUR in 2011 and it has experienced annual two-digit growth rates for the past ten years. The company has a total of approximately 1700 employees – 1400 in Denmark and the remaining 300 in offshore locations, including China, Switzerland, the Czech Republic, and the Philippines. GlobalSoft is largely involved with developing public and governmental systems for the Danish society. They develop highly specialized solutions for public institutions and private companies located primarily in Denmark. Thus a large part of the work requires close collaboration with representatives from the Danish government. While GlobalSoft originated as a solely Danish company, it has expanded across the Danish borders to include offices in the Philippines, China, and various European countries.

In 2009 GlobalSoft acquired the Philippine supplier, which at the time had approximately 85 employees. The Philippine company had a Danish CEO with strong ties to GlobalSoft and it had originally operated as an independent supplier of programming resources to GlobalSoft.

2.1.1. The DAFIC project

The purpose of the DAFIC project was to create efficient and easy access to public data in the Danish public sector by establishing a new access infrastructure to the public service databases. The new infrastructure system allows citizens and companies to gain access to all public websites with only one set of login credentials, making it easier to work across multiple public platforms. The project involves not only Danish citizens but also more than a hundred external service providers, who also use the same data. The project had a high level of technical and organizational complexity since the DAFIC system had to be technically connected to a large range of electronic systems already in use. In addition, the project got off to a really bad start because of an overly optimistic contract design. Despite these factors, the project is – although delayed – now completed and successfully implemented in Danish society.

2.1.2. The Danish Office

The Danish employees in the DAFIC project were initially located in an open office space in a suburb of Copenhagen but were relocated several times during the three years. Many of the Danish employees had been working in GlobalSoft for at least five years or more and many still recalled the time when GlobalSoft was only operating from Denmark. 15 full-time employees were working on the DAFIC project as well as a couple of part-time workers. The age distribution ranged from early thirties to late forties with an average of mid-forties. The group consisted of three IT architects, a project leader, a test manager and two external test consultants as well as a test trainee. The remaining seven employees were developers. It should be noted that these numbers are estimates as the project used numerous people to a varying degree during the three years. The employees in the Danish office were responsible for scoping, writing the requirement specification and assigning tasks. They were also solely in charge of handling client relations. The client consisted of a group of five to six people, including a project leader and technical experts. They were present in the Danish office once a week, participating in meetings and working with the developers. The employees in the DAFIC project also collaborated with another office in Denmark, namely the back office group. This group of people were not very visible in day-to-day interactions as they were situated at another location. However, towards the end of the project the back office team was collocated with the Danish developers. The back office group was responsible for the technical implementation of the software and also for delivering the build and test environments. They had no direct collaboration with the Philippine employees and were thus not the primary focus of this research.

2.1.3. The Philippine Office

The Philippine employees in the DAFIC project were working in a large open office space that was shared with the rest of the employees in the Philippine office. The DAFIC project employed 12 to 15 people, including a project leader, a system

analyst, two testers and a group of developers. The age distribution ranged from mid-twenties to late forties with an average of early thirties. Thus they were younger than the Danish employees and were typically hired directly from universities. They were responsible for developing software, testing and writing a part of the product descriptions. They were working in a strict time-managing system where they would have to sign-in during the morning and to sign-out whenever they wanted to leave the office, including bathroom breaks and lunch. They had a high proficiency in English due to the fact that the Philippines had been an American colony for 50 years.

2.2. The Data Sources

Three researchers from NexGSD (including the author of this dissertation) and a research assistant were involved in collecting data in GlobalSoft. However, only the author of this dissertation was following the entire DAFIC project, which includes the vast majority of the data presented here (see Table 1). The data collection began with a preliminary interview process that was initiated in November 2010 and ended in May 2011. After the interview process, the author of this dissertation followed the DAFIC project over time. The data collection in the DAFIC project consisted primarily of interviews and observations of the work practices in both Denmark and the Philippines. In total, we conducted 28 audio recorded interviews (19 in Denmark, 9 in the Philippines) with an average length of 50 minutes. During this time we also conducted and video recorded four workshops (two in Denmark and two in the Philippines) for further analysis. Employees at many different organizational levels were interviewed, allowing us to compare perceptions of the corporate vice-president of GSD with those of the developers.

The majority of the data collected consisted of observations of the DAFIC project made by the author of this dissertation. Observations were made in Manila, Philippines, for approximately five weeks during two separate phases (November 2011, and January 2012). During this time the researcher spent four to five days a week in the Philippine office. Observations in the Copenhagen office spanned a period of 12 months where the first four months were the most intensive. During the first four months, the researcher spent three to four days a week in the DAFIC project. During the remaining eight months, the researcher returned approximately every other week on average. The observations focused mainly on the cooperative work practices among globally distributed project members. During the data generation process, the initial empirical findings were discussed with the DAFIC employees, either during casual conversations or during official presentations. Four official presentations (three in Denmark and one in the Philippines) were given to communicate the findings with the company representatives during the fieldwork period. The presentations gave the practitioners an opportunity to give feedback by discussing and providing additional information to particular observations. The feedback from employees was noted and included in the different stages of analysis. Feedback sessions were used to challenge and validate our findings. The author also examined internal company documents, requirement specifications, and official materials. Finally, we did screen recordings of four employees (two located in Denmark and two in the Philippines) for two complete working days. The combination of data collection techniques complements each other well when pursuing a deeper understanding of a particular work practice (Blomberg & Giacomi 1993). Pursuing an in-depth understanding of global software teams

requires a breadth and plurality in methodological choices to comprehend the complexity of the field of study (Patil et al. 2011).

Table 1: The Data Sources

Time & location	Data sources	Use of data in analysis
November 2010, Denmark	14 preliminary audio recorded and transcribed interviews with employees at GlobalSoft, ranging from developers, IT architects and managers, to the vice president <ul style="list-style-type: none"> - Interviews lasting 30–60 minutes - Average length of 50 minutes 	To establish an understanding of the company and the project structure in global software development To identify key challenges of the collaboration between Denmark and the Philippines
December 2010 – January 2011, Philippines	Two interviews with the manager and the project leader Observation of the everyday practices Presentation to and discussion of initial findings with DAFIC members	To achieve an initial understanding of the Philippine perspective
Spring 2011, Denmark and the Philippines	Four workshops conducted – two in Denmark and two in the Philippines	More detailed knowledge of the Danish and Philippine perspectives on the collaboration
July 2011, Philippines	Three interviews with the team leader, tester, and manager First workshop on opportunities and challenges in GSD Ten 8-hour days of observations, including observing developers, project managers, and testers. Different types of meetings were observed.	Further understanding of the collaboration from the Philippine perspective
October – December 2011, Denmark	Observations of the DAFIC project 2–3 times a week lasting between four to ten hours <ul style="list-style-type: none"> - Observing work practices - Observing team meetings, steering meetings, and client meetings - Informal audio recorded interviews lasting five to ten minutes 	To understand the everyday practices and challenges of global software development in DAFIC from a Danish perspective
January 2012, Philippines	Observations of DAFIC four times a week lasting between four to ten hours <ul style="list-style-type: none"> - Observing work practices - Observing team meetings - Informal audio recorded interviews lasting five to ten minutes Four audio recorded interviews with manager, project leader, project manager, and system analyst <ul style="list-style-type: none"> - Interviews lasting 30–60 minutes - Average length of 45 minutes Audio recorded presentation of initial findings for management in the Philippines lasting 60 minutes including validating feedback. Creating screen recordings in both DK	To understand the everyday practices and challenges of global software development in DAFIC from a Philippine perspective

	and PH.	
March 2012 – October 2013, Denmark	<p>Five audio recorded and transcribed follow-up interviews and observations in DAFIC (average length of 50 minutes.)</p> <ul style="list-style-type: none"> - Monthly follow-up observations - Attending evaluation meetings for DAFIC project - Questionnaire sent to the Philippine employees for follow-up <p>Presentation of findings for the DAFIC team</p> <p>Presentation of high-level implications to the GlobalSoft management group</p>	Validating interpretation and understanding of the collected empirical data

2.3. The Analytical Approach

Grounded theory originally proposed that the researcher should enter the field with an open mind letting the data emerge from the field and leaving all theoretical assumptions behind (Strauss & Glaser 1967). Moreover, grounded theory requires the researcher to “empty” the field from relevant data involving a process of entering and re-entering the field numerous times until the field can be claimed to provide no new insights. This process is time-consuming and very demanding for one researcher and very difficult in an ever-changing field such as the DAFIC project. Thus I was not able or willing to commit to a full-hearted grounded theory approach. Instead I entered the field with an open mind but was also guided by a specific focus on challenges and opportunities in the collaboration across a Danish and Philippine software development office. I had decided to focus on work practices from a CSCW perspective and was therefore guided by certain theoretical assumptions.

Grounded theory is also an analytical process that offers a systematized approach for developing insights from vast amounts of empirical data. The use of open coding is part of a grounded theory approach and is described as a process where data can be separated and organized and compared (Strauss & Corbin 1990). The researchers look through all the data and develop categories. Initially, these categories will be crude and general but later in the process these categories are developed and refined. For example, the category “culture” can be further refined into sub-categories such as “national culture”, “organizational culture”, “individual culture”, etc. Coding of data can be described as an iterative process where the categories are formed and interpreted by the researcher and is thus in “a constant state of potential revision and fluidity” (Bryman 2004, p.402). The purpose of coding is to apply a systematic approach to the analysis to ensure that findings span numerous events that form certain patterns of behaviour or activities. Typically, the coding of data spans numerous stages where categories are formed and revised among several researchers.

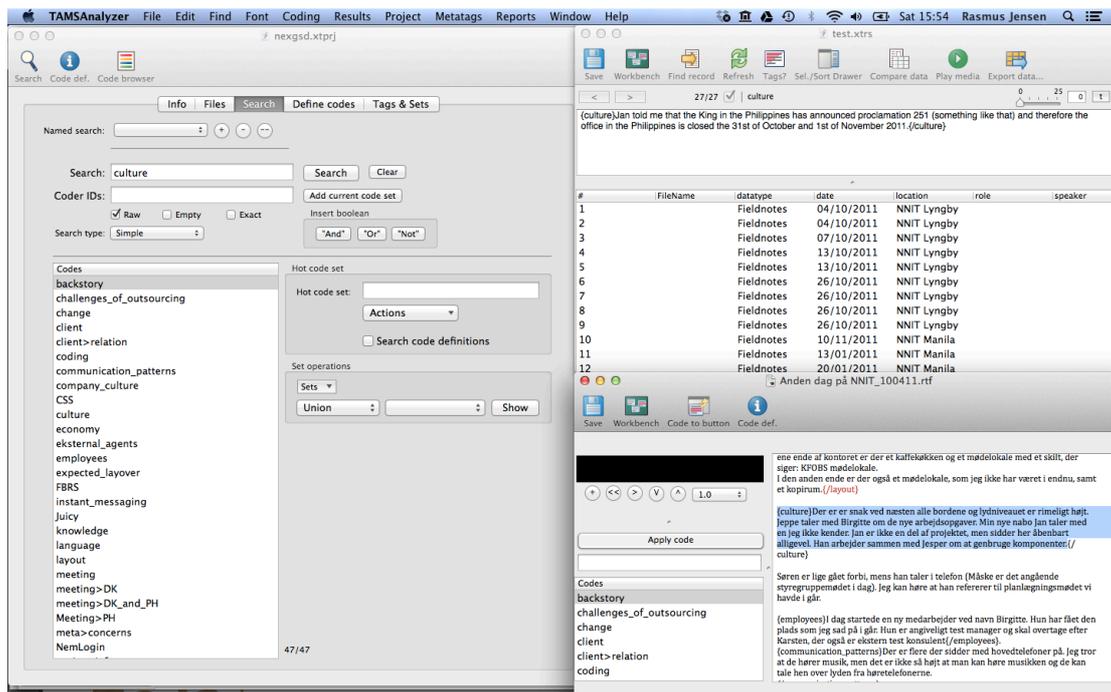
My initial coding process involved a non-technical method for using colour coding in a Word document (Figure 1). It was a tedious and not very practical approach to coding large amounts of data.

Figure 1: The Data Analysis



As the amount of data grew, I had to reorganize all the data into qualitative data analysis tool to be able to manage and organize it all. Many different tools exist but I chose the TAMS Analyser because it was easily available and operated on a Mac computer (Figure 2). While the design of this tool was not as professional as Nvivo or other similar tools it worked perfectly for my needs.

Figure 2: The Data Analysis



The TAMS Analyzer allowed me to search across multiple categories providing easy access to renaming and revising individual categories and easily merge categories. The coding process is an essential part of the analysis since it allows the researcher to systematically organize the data. My own coding began with the formation of general categories that emerged from the data extracts. I would create a category whenever I found a data example that was relevant for my initial research question. A category

such as “language” could be related to a challenge in the collaboration, so whenever language came up in the data material, I would categorize it to the “language” label.

After this first level of coding, the analysis became more refined. Some categories were changed, divided into smaller categories or simply renamed to match the content of the category better. The coding and the categorization of data allowed patterns of behaviour to emerge that guided the research questions for the papers. It was an iterative process that included more than one researcher to validate and discuss the categories and ensuring that the data examples actually represented a general pattern of behaviour in the team. As an example, in Paper No. 1 we ask: *How can we identify situations where the differences in social worlds between geographically distributed developers become salient in their everyday interactions?*

This research question emerged by identifying all the data material that was referring to culture and analysing how the practitioners described situations or activities in relation to culture. The process of identifying patterns in the data material was a process of *reflection* among several researchers which is a critical part of doing ethnography. Doing ethnography is however more than a way of doing things. Ethnography as fieldwork involves considerations of the object of the research namely to study people in naturally occurring settings. The observer is not a neutral entity but a natural part of what is being observed. The researcher is a participant in the setting and is required to reflect on his or her role in that setting to avoid obtrusive or imposing behaviour (Brewer 2000). Observations in a naturally occurring setting with meaning-endowing people consequently mean that data is contextually linked to the context and location of the study and the methods used for collecting the data (Brewer 2000). Critics would argue that this leads to a partial and selective data collection process, which is also true and leads to the challenges of representation and legitimation. The challenge of representation and legitimation of the findings is present for all scientific research and is addressed in different ways. Ethnographers have taken on this challenge of validity, reliability and generalizability by turning to reflection (Brewer 2000). A reflective approach involves a critical attitude towards the data collected, which includes reflection on possible power relations in the study, the context and location of the study as well as personal biases. *The main meaning of reflexivity is that the scientific observer is part and parcel of the setting, context and culture he or she is trying to understand and represent* (Altheide & Johnson 1998, p.285). Reflexivity involves reflection by ethnographers on the social processes that impinge upon and influence data. It requires a critical attitude towards data, and recognition of the influence on the research of such factors as the location of the setting, the sensitivity of the topic, power relations in the field and the nature of the social interaction between the researcher and the researched, all of which influences how the data is interpreted and conveyed in writing up the results (Brewer 2000). The analysis of the data using a critical attitude towards the findings and reflecting on the context of the data allowed us to develop deep insights into the work practices of global software developers.

2.4. The Role as a Researcher in an Ethnographic Workplace Study

My focus in the field was the collaboration between the Danish and Philippine offices and I was specifically looking for interactions or work practices between globally distributed practitioners in DAFIC project. It is important to clarify the role of the researcher in order to understand this process fully: 'Interpretive researchers are not saying to the reader that they are reporting facts; instead they are reporting their interpretations' (Walsham 1995, p.70). While in the field the research question guided an open-ended approach to the practices observed. Thus I was not 'prescribed' by any preordained theoretical interests but rather tried to let the field inform me as I went along (Randall et al. 2007). During the early stages of the fieldwork, time was spent meeting the project members, noting down the office layout and getting a feel of the everyday routines. In later stages I began to 'shadow' key employees in the project, i.e. I observed the employees from morning to evening. I also attended a range of meetings with both clients and other project participants. The same procedure was pursued in the Philippine department. Daily practices and routines as well as surprising events were all noted down. After a day of observations the notes were rewritten and logged for later reflection. Interesting and surprising observations were discussed with the other researchers in between observations. This process helped to pinpoint central themes in the fieldwork. Data was not 'collected' from the field. Instead note-taking and observations should be seen as a selection of events that, in the eyes of the observer, were relevant to write down. According to Walsham (Walsham 1995) it is key to have an open-ended research methodology without ignoring previous work done within the field of study. The preliminary round of interviews gave a basic understanding of the challenges and the fieldwork would allow for investigating them in greater detail. This background knowledge helped to guide what to look for and take note of while in the field. However, at the same time I strived at keeping an 'open mind' and leave my assumptions of the collaboration behind me to allow for a deeper understanding of the rationale behind people's actions in the collaboration (Nardi 2010).

I was constantly trying to observe interesting events and practices, attempting to gather as much data as possible. I have written hundreds of pages with observations, notes and insights. A day of observing would easily be followed by four hours of writing down the observations and refining my hastily scribbled notes. It was important to get this work done within a 24 hour cycle so I would not forget the little details. I was not always sure what to do with all the data I accumulated and most of it has not been used directly. However, it helped to establish a detailed picture and provide 'thick descriptions' of the work practices involved in global software development (Harper 2000). On the side I created documents where I wrote down questions or episodes that needed further exploration as part of the on-going analysis of the data. Some of these episodes were also shared with my supervisor and other colleagues in order to identify patterns in the data.

2.5. Reflections on my Role as an Observer in the DAFIC Project

During the first two months of my observations, a major challenge for me was actually trying to understand what the practitioners were talking about. The complexity of the project was immense and the terminology and technical terms were foreign to me. I spend a lot of time trying to decipher the 'code' by writing long lists

of words that I did not understand. In time things became easier and I was able to make connections between previous events and make references to technical aspects when engaging with the practitioners. I also spend considerable amounts of time trying to map out the project in terms of people involved by drawing sketches of the offices in both Denmark and the Philippines. However, from a CSCW perspective the main issue with ethnography is not about “going native” but rather to ensure that the views of the researcher are treated respectfully by those observed (Harper 2000). When I first entered the project as an ethnographer, my initial objective was to make a good impression. I wanted people to like me and worked hard on establishing a relationship with the people, hoping that they would explain and show me how they worked. My rationale behind this was that I wanted to be in a position where people would feel safe about sharing sensible information and would not have to worry about my presence. As Randall et al (2007) explains: ‘Ethnographic practice is simply about presenting oneself as a reasonable, courteous and unthreatening human being who is interested in what people do and then shutting up, watching and listening’ (p. 181). To achieve this I took every opportunity offered to spend time with the practitioners both at and outside work. For example, I participated in a basketball match and sang karaoke with people from the office in the Philippines. In Denmark I regularly had lunch with the group and arranged to join some of them on a trip to the Philippine office. Despite my best intentions, it did not always go as planned especially during the first few weeks of observing. For instance my very first chance to meet the entire group in the Danish office was thwarted by a misunderstanding. Below is an extract of my observations at that time.

12.20: I have been invited for lunch with the project members that should start at 12.00. However, I cannot find the group and I tried the other restaurants in the area. None of the waiters have heard of the event. It is indeed an awkward feeling. I have tried calling the project leader and I also texted him but I cannot get hold of him. I also tried calling Pernille (my supervisor) just to get some sort of reassurance but she did not answer either. I am unsure of what to do know. I am currently sitting at the café where we were supposed to meet and I guess I will wait until 13.00. I left the café at 13.30 without seeing anyone from the project. On the way home, I get a call from the project leader and he said that they were all sitting at the harbour, where the café also had a room. I did not know about this place and the waiters had not informed me about this other location. (Observation, Denmark, 29-09-2011)

Certainly, my first opportunity to meet the Danish group of developers did not go as planned and I felt terrible at the time. Not showing up for a very gracious invitation did not exactly establish the kind of relationship I had in mind. Reflecting on my initial experiences in the DAFIC project, I realized that achieving familiarity with the team was not enough to ensure that my observations were to be treated as from the “inside” rather than as from the “outside”. These reflections have led me to believe that both the Danish and Philippine offices offered specific challenges and I had to undertake specific activities to overcome these. These challenges can be referred to as “ritual inductions”, i.e. ritual transformations of identity are important to ensure that findings emerging from the ethnography can be taken seriously (Harper 2000). It is about getting the “inside” perspective of the area of research but ritual inductions can be comprised of many different things and sometimes a certain amount of time is all that is required. However, in my case I would point to two specific activities that can be regarded as a ritual induction. The practitioners in the Danish office quickly

understood and welcomed my role in the team. We shared a similar background and language and there were few hierarchical structures in the team unlike in the Philippine office. Even so I felt that I did not enter the “inner circle” before I went to visit the Philippine office. Going there was to the Danish practitioners an indication of my willingness to know what was actually going on. It was an important ritual induction on the Danish side because it allowed me to share anecdotes with other Danish team members who had also travelled to the Philippine office and experienced it. Returning from my first travel, I felt that I had earned some level of trust in the Danish office. The situation was as follows: I was attending an informal meeting between the Danish project leader and the chief IT architect and they were discussing the work efforts of an external consultant in the project. At some point the chief IT architect said something like ‘Well, between you and me...’ and then continued to discuss a personal issue about one of the other employees in the project. The fact that the project leader was willing to share sensitive information in my presence made me realize that the IT developers had grown accustomed to me.

The Philippine office was an entirely different experience and I had to engage in other activities to reach the same level of trust. I could not rely on a shared background or understanding of language and they had a more formalized hierarchical structure in the organization. Initially, I was seriously concerned about being seen as a representative of GlobalSoft, secretly trying to evaluate work performance among the Philippine team members. Nevertheless, I found it relatively easy to connect with the Philippine employees, maybe because we shared a range of similar interests. Focusing on our shared interests in the Philippine office was part of my strategy of not being seen as corporate person. However, the key activity that led to “inside” knowledge was stressing the fact that I wanted to represent their voice in matter of challenges in the global work. I did this by making a presentation of my initial findings for employees in the Philippine office and having an open discussion about their perspectives on global work. Thus I was making myself a mouthpiece for their concerns paving the way for the ‘ritual transformation of identity’ described in the literature (Harper 2000). I was no longer seen as a corporate representative but a person who could push certain pressing agendas for the Philippine employees. This transformation helped me to get better access to the data material in the field.

3. INVESTIGATING GSD FROM A CSCW PERSPECTIVE

The area of global software development has increasingly become a field of interest for researchers within the field of CSCW (Herbsleb et al. 2000; Espinosa et al. 2003; Steinmacher et al. 2013; Matthiessen et al. 2014). CSCW research focuses on collaborative work and the nature of the task. Collaborative work is defined as a cooperative arrangement where several people are mutually engaged within a common field of work, and where individual activities have a direct impact on the collaborative partners (Schmidt & Bannon 1992). Investigating GSD from a CSCW perspective gives the researcher an opportunity to understand the detailed aspects of collaborative work practices by using theoretical concepts from CSCW research. Work practices of global software development teams include the contractual setup, bidding phase, requirement specification and planning as well as the actual development, documentation, testing, deployment, maintenance and support. CSCW concepts provide a framework for identifying and describing the work practices of GSD. Research has showed that the orchestration of global software development requires establishing communication patterns, information sharing and creating a common history (Herbsleb et al. 2005). In particular, researchers have found that these activities require strategies for handling articulation work that are different from collocated software development projects. Previous research has emphasized the importance of articulation work (Boden et al. 2008), knowledge sharing (Avram et al. 2009), coordination (Redmiles et al. 2007) and culture (van Marrewijk 2010) in global software development projects. However, there is still have a fairly limited understanding of the intricate interactions among different practices in the everyday work of global software teams (Herbsleb 2007). Using this as a starting point, the research question investigates how IT developers coordinate their everyday work practices to adapt to recurrent changes in global software development projects.

As a researcher within the field of CSCW, the focus is specifically on the work practices and how discontinuities affected the work in terms of the extra articulation work involved in solving the task. Consequently, this section will present previous CSCW research on strategies for reducing the articulation work and specifically on strategies of coordination and establishing awareness in a GSD setting. Spanning geographical distances and cultural differences creates demanding conditions for establishing a shared context or common ground (Gary M Olson & Judith S Olson 2000; Judy S Olson & Gary Olson 2013). Thus this section will also present previous research as well as the role of culture in global software teams. The following sections present major findings of the CSCW research – mostly from cross-sectional studies of GSD projects - and relate these findings to the way distributed team members adjust to discontinuities over time in collaborative work practices in a global software development company. The following sections will focus on the existing research on awareness and coordination strategies as well as the impact of cultural differences to provide a theoretical framework for the main content of the four papers. The rationale behind this structure is to establish a framework for each of the four papers and present “gaps” in the literature that will be addressed in more detail in the papers. Thus sections 3.1 and 3.2 present the setting and premises of articulation work. Sections 3.3, 3.4, and 3.5 are specifically designed to provide a theoretical framework for the main part in terms of the papers and each of these sections are concluded with

a selection of questions. These questions will be addressed in the papers as well as in the discussion.

3.1. Understanding Globally Distributed Work

The field of global software development research have steadily increased over the past decades (Carmel 1999; Prikladnicki et al. 2003; Carmel & Tjia 2005; Herbsleb et al. 2005). During this time, outsourcing and offshoring of software development has gained an increased interest for companies due to advances in communication technologies and the rise of emergent economies, such as India and China. The IT industry was one of the pioneers in outsourcing starting with companies such as IBM in the early 1980s (Lonsdale and Cox 2000). The advantages and disadvantages of offshoring are still being discussed (Aspray 2006; Šmite et al. 2010). Some embrace the cosmopolitan aspects of a global work force whereas others argue that offshoring is a threat to the local work force and thus should be prevented, for example in media reports describing the consequences of offshoring, such as loss of jobs, know-how, and innovation (Pedersen 2012). On the other hand others promote the merits of offshoring with examples of successful companies (Pedersen 2011). The distinction between ‘offshoring’ and ‘outsourcing’ is worth mentioning although global software development studies engage in both types. ‘Offshoring’ refers to solving a task within the same company but at another location, typically in another country. ‘Offshoring’ originally meant ‘crossing a shore’ for instance in US companies that were offshoring to Asian vendors. However, this is not a prerequisite for an offshore relationship. Today ‘offshoring’ happens in many parts of the world and between different nationalities (Conchúir et al. 2009). ‘Outsourcing’ on the other hand typically refers to having another company solve a task for a company and this is regardless of the location of that other company. In this study of an offshore software development team, the term ‘offshore’ refers to tasks assigned by employees in a company located in one country to employees in the same company located in another country.

The research field of GSD concerns industry studies of both major international corporations (Søderberg et al. 2013), smaller development companies (Boden et al. 2008; Esbensen & Bjørn 2014), open source development (Irani & Silberman 2013) and to some extent developer ecosystems such as Apple’s iTunes Store (Bergvall-Kåreborn & Howcroft 2013). The common denominator for research in the GSD field is to investigate software development from a global perspective. Global seems to imply multiple sites across the world but many studies of global software development are merely cross-national (van Marrewijk 2010; Søderberg et al. 2013; Montoya-Weiss et al. 2001). A recent literature review on global software engineering found that a majority of the studies involved intra-organizational collaboration across two sites (Šmite et al. 2010). Thus the study of global software development implies an investigation of software development in teams distributed across two or more geographically dispersed locations spanning different time zones and nationalities. Researchers in this particular area have focused their efforts on developing technical solutions (de Souza et al. 2007) to aid the practitioners, propose best practices (Prikladnicki et al. 2003), and create a deeper understanding of the intricacies of work practices (Avram et al. 2009; Matthiessen et al. 2014). Researchers engage in understanding the everyday work practices and provide rich insights to the field research, such as establishing trust (Al-Ani & Redmiles 2009) or creating awareness (Boden et al. 2014). Consequently, these studies focus not only on a technological fix

but also aims at identifying how practices of coordination, knowledge and awareness strategies unfold in a globally distributed setting.

3.2. The Extra Effort Required to Solve the Task

The fundamental assumption of cooperative work, is that it is inherently distributed because several actors are involved and furthermore that, in complex settings, the articulation work does not happen automatically (Schmidt & Simone 1996). In collaborative work settings where people are mutually dependent on each other extra work is required to coordinate the work between the individuals (Schmidt 2011). In CSCW research, the extra work required to solve the task, has been referred to as ‘articulation work’ and it is a key concept for understanding the interwoven practices of interaction between actors working on a mutually shared task (Blomberg & Karasti 2013). In a global software development setting, we can use articulation work to become more specific about what collaborative work really entails. Articulation work can loosely be defined, as all the extra work required for performing a collaborative task. The concept of articulation work was developed by Anselm Strauss (1988) in his attempt to describe the processes of coordination in projects. According to Strauss (1988), articulation work refers to ‘the specifics of putting together tasks, task sequences, task clusters – even aligning larger units such as lines of work and subprojects – in the service of work flow’ (p. 168).

A classic example of articulation work could be the task of moving a piano. The work itself can be described as moving the piano from one point to another. The articulation work is all the extra work required for two people to actually move the piano. That work consist of both verbal and non-verbal ways of communication, for instance planning the route and the way of lifting the piano but also the gentle movements that helps both people to maintain balance and direction. In this very simple situation of cooperative work, articulation work is required to coordinate the movements between the two people. However, working in a globally distributed setting greatly increases the complexity of the collaborative tasks and it is no longer as straightforward as where and how to perform the required articulation work. When people can no longer see each other or may not even speak the same language the articulation work becomes much more complicated. Moreover, complexity increases when the task is developing new software tools rather than moving a piano. Thus the effort required to solve the task increases and the practitioners must develop strategies to manage all the extra work required to perform the task.

Research has shown that the time and the communications required to perform the same site software development increases greatly when practitioners have to work across different sites (Herbsleb et al. 2001). CSCW research has engaged in identifying and supporting different types of strategies for reducing articulation work by investigating coordination and awareness strategies and knowledge practices (Jarvenpaa and Leidner 1999; Hinds and Kiesler 2002; Larsson 2003; Massey et al. 2003 Gibson and Gibbs 2006; Bjørn and Christensen 2011). One of the major challenges of articulation work in globally distributed teams, is making activities visible so others can act upon them. The next section will take a closer look at how CSCW researchers describe the process of making things visible in globally distributed collaborative work settings.

3.3. Making Work Practices Visible

In globally distributed teams, it is vital for the team members to understand what the other team members are doing (Herbsleb 2007). In part this can be achieved through practices of making essential aspects of the work visible to other people in the collaboration and vice versa (de Souza & Redmiles 2011). Visibility in the collaborative work allows team members to adjust their own work or helps to coordinate tasks. Practices of displaying and monitoring in collaborative work are thus essential for seamless and effortless coordination and within CSCW research these practices are referred to as *awareness* (Gross 2013). The main agendas of CSCW is to investigate the basic nature of collaborative practices with the aim of designing collaborative systems (Schmidt & Bannon 1992) and awareness studies have played an important role for this agenda, for example by studying the basic nature of air traffic control (Bentley et al. 1992), or creating awareness supporting tools (Gutwin et al. 2004).

In globally distributed teams, awareness does not happen automatically and has to be supported by continuous communication (Gutwin et al. 2004), which has been the point of departure for awareness studies of globally distributed teams. However, while previous research has focused mainly on supporting the displaying and monitoring of core programming tasks using tools (Steinmacher et al. 2013), there has been little research on why displaying and monitoring are important for demonstrating commitment in globally distributed teams (Jarvenpaa & Leidner 1999). Looking back on the CSCW research, it is clear that awareness has been a core interest to CSCW researchers, but primarily in collocated settings (Steinmacher et al. 2013). Awareness consists of two interlinked practices that, according to Souza and Redmiles, ‘involves (i) displaying one’s actions, and (ii) monitoring others’ actions’. That is to say, social actors monitor their colleagues’ actions in order to understand how these actions impact their own work and, while doing their work, social actors display their actions in such a way that others can easily monitor them’ (de Souza & Redmiles 2011, p.2).

This notion of awareness emerged during studies of practitioners in collocated settings to explain how skilled practitioners coordinated effortlessly and seamlessly between them (Bentley et al. 1992; Heath & Luff 1992; Ackerman et al. 2013). From these studies we learned that coordination in a collocated setting involves an intricate process of displaying and monitoring certain activities. Thus awareness is a process requiring relevant information about your own work as well your surroundings’ in order to display and interpret what others are displaying and act accordingly. A classic example that illustrates the research on awareness in CSCW, is the early studies of Heath and Luff (1992) that showed how skilled subway controllers engaged in seamless and unobtrusive actions by displaying specific information to colleagues or anticipating other colleagues’ activities by monitoring their work. Similarly, a study of air traffic control demonstrated the importance of flight strips and their particular alignments on the process board (Bentley et al. 1992). The authors argued that the particular way of organizing the flight strips, provided information about the state of the air space, which was made visibly available for others who passed by and allowed them to act accordingly (Bentley et al. 1992). Thus such awareness refers to the information about the activities of other group members (Steinfeld et al. 1999; Schmidt 2002). Awareness was identified in these early studies as the effortless practice of aligning distributed and yet interdependent activities by highly skilled

practitioners enacted in particular artefacts. However, as it turned out, ‘such effortless activities of alignment’ can be interpreted in various ways and thus mean different things in different studies (Schmidt 2002).

Moving on to studies of awareness in globally software development teams, research has proposed technical solutions in which practitioners could make the work visibly available for others to monitor and act accordingly (Steinmacher et al. 2013). Awareness features like ‘online/offline’ indicators available in most modern communication software such as instant messenger or email are but one example of how to display and monitor others. Recent research have proposed tools to support a better understanding of *who* is working in the project and what shared dependencies people may have. One such tool is ‘Adriane (de Souza et al. 2007). The aim of Adriane was to identify technical dependencies between one or more developers and visualize these in a social network diagram. Using the tool, developers can become more aware of the interconnectedness between them and identify choke points in the development phase. Moreover, it enabled a visualization of the technical dependencies in the software which otherwise is invisible and achieved this in a continuous and effortless way (de Souza et al. 2007). Knowing *what* people are working on is another central aspect of awareness and when people cannot see each other, this information has to be conveyed through other means. One way to address this aspect has been to improve the informal communication in the workspace. Fitzpatrick et al. (2006) proposed a tool to enable chat and informal discussions in the code management system. This enabled the developers to coordinate and manage the different tasks and over time they developed a team culture. Lastly, the *when* and *where* aspects of awareness in global software development teams have been supported for example by visualization techniques and code annotation tools (Steinmacher et al. 2013). The former is illustrated in the tool ‘Palantír’ that enabled developers to see changes that other developers implemented in parallel to their own work. This kind of awareness allowed the developers to react early to conflicts in the code and avoid breakdowns in the code at later stages where they are more costly (Sarma and Hoek 2002). The latter is exemplified by code annotation that shows how awareness can be embedded in shared artifacts (Steinmacher 2013). Dekel and Herbsleb (2008) enabled developers to write informal comments and share them with other developers. Thus the tool allowed the community generated knowledge to be directly linked to the relevant source code (Dekel & Herbsleb 2008). Each of these studies represents different aspects (such as the *who*, *what* and *where*) of making work practices visibly available for team members of globally distributed teams so they can act accordingly and coordinate their efforts.

The CSCW literature on awareness in global software teams explains the importance of enabling practitioners to display and monitor activities and act accordingly. Consequently, CSCW researchers have proposed a range of design solutions both to collocated and distributed settings. While these studies use different approaches (and artefacts) to support awareness in their respective settings there is little emphasis on the skills and efforts required to make the displaying and monitoring relevant to the practitioners. However, supporting awareness in globally distributed teams involve not only specific tools but more importantly an understanding of the intricate work practices used when practitioners engage in global software development. A strict focus on providing tools for specific tasks and systems seems to neglect the importance of establishing a coherent picture of software development as a sum of

processes, for instance the underlying premise of shared meaning that is required for awareness. Thus it is not enough to simply remedy the lack of awareness at the workplace by implementing ‘awareness’ features to specific systems.

While studies that support separate tasks are important, the tendency to focus on tools supporting awareness when doing particular tasks in globally distributed teams, it is also necessary to understand what it is that links these practices together. Awareness in global software teams is consequently linked to practitioners’ ability to understand and convey the visible information available in the teams. Consequently, establishing mutually shared attention in global collaborative teams requires more than ‘awareness of some technically constructed environments where individuals are working together’ (Leinonen et al. 2005, p.316). Globally distributed work consists of a range of interlinked tasks where the technical aspects are only part of the work (Handel and Herbsleb 2002). Other important tasks include planning the project, running meetings, documentation and informal talk that together shape the shared meaning of the project among the participants. For example, previous studies show that more than 15% of the collaborative work consists of informal talk (Grinter et al. 1999) supporting the importance of informal practices. Thus we need a better understanding of how awareness and attention facilitate not only coordination among practitioners but also other aspects. For instance we know that showing attention is related to commitment in globally distributed teams (Jarvenpaa & Leidner 1999; Nardi 2005)) but only little of the link between practices of tracking attention and the demonstration of commitment in a globally distributed setting. How do practitioners of global software development teams use attention to establish and maintain commitment? How do practitioners track each other’s commitment in software projects spanning different geographical locations? What are the practices of tracking, interpreting and aligning attention using the information available in the tools? While these questions will not be directly addressed in the discussion they are central for framework of Paper No. 3 and are thus included here.

3.4. Coordination Practices

In globally distributed teams, the efforts involved in handling the articulation work can become overwhelming for the practitioners. In these situations the teams rely heavily on coordination strategies to reduce the efforts of articulation work. Coordination is about managing dependencies and one of the key challenges in software development is to ‘unravel the complex relationship between software dependencies and task dependencies’ (Herbsleb 2007). Coordination in terms of planning, scheduling, and the allocation of resources helps to coordinate activities in projects where several people work closely together (Montoya-Weiss et al. 2001). In the field of CSCW, researchers have identified certain characteristics about the coordination mechanisms that practitioners use to support the articulation of cooperative work (Schmidt & Simone 1996). Coordination mechanisms are defined as a specific organizational construct that has a coordinative protocol that is imprinted in a distinct artefact. Research has found that these coordination mechanisms can have the ‘weak’ role of a map, such as a project plan that merely guides the practitioners (Suchman 1995). Moreover, coordination mechanisms can facilitate the ‘strong’ role of a script that ‘offers a pre-computation of interdependencies’ such as a checklist (Schmidt & Simone 1996). Coordination mechanisms allow computational support for cooperative work, which is useful for handling the articulation work. While

coordinative artefacts play a role in global software development in terms of project plans, product descriptions, and mock-ups, then another essential aspect of coordination materializes in the division of tasks in GSD projects.

Collocated software development requires a lot of coordination (Grinter 2003) and geographical distance only increases the need to coordinate (Boden & Avram 2009, Avram et al. 2009). In global software development projects the IT developers are often dependent on each other in order to solve the task especially in those situations where the final product is applied into a specific domain area, for instance when one group of IT developers has the necessary domain knowledge, which needs to be communicated across the geographical distance (Espinosa et al. 2007). Large software products are typically decomposed into smaller products and then divided between developers to enable a faster process (Grinter 2003). The rationale behind this is based on the challenge of working across a geographical distance and possible cultural differences. Thus to minimize interactions across locations, the software product is decomposed into smaller packages and developed on location (Hertzum & Pries-Heje 2011). Nevertheless, coordination is required during the reassembling of the software to ensure that dependencies between software packages are correct. This is especially true for complex software development tasks that share many dependencies across the individual products and thus practitioners can *rationalize* this process by using coordination mechanisms (Gerson 2008). This can be done by standardization to make connections and relationships between things uniform and thus easier to handle. Another coordination strategy is segregation, which is typically used in the domain of software development to separate tasks and remove dependencies among them (Gerson 2008).

Global software teams engage in a range of coordination practices to manage the dependencies between the different software components. This work requires both formal and informal coordination that is often technology mediated, as the practitioners are rarely able to meet. Practitioners use documents, project plans, test management systems, coding data bases, virtual build environments as well as email, video meetings and chat to coordinate their tasks. Research of coordination in global software development teams has suggested many individual solutions both technical and non-technical but has revealed little about the effects of the integration among them (Herbsleb 2007). Moreover, research even suggests that the need for more coordinative tools in GSD is not supported (Boden et al. 2007; Whittaker & Schwarz 1999; Leinonen et al. 2005). An argument against a sole focus on technical solutions is that articulation work is a continuous effort especially in small agile software development teams (Boden et al. 2007) and coordination mechanisms such as project plans are neither current, accurate or credible (Whittaker & Schwarz 1999). This is not to say that coordination technologies are not important but that practitioners already have tools for coordination available. The main problem according to these studies is that research needs to refocus its efforts on investigating how to facilitate informal patterns of communication (Boden et al. 2007). Consequently, it seems that the strong emphasis on managing individual dependencies with tools has left a gap in the literature regarding the complexity of practitioners' interactions in globally distributed teams. For example, one study investigated the role of temporal coordination mechanisms for conflict management in globally distributed teams (Montoya-Weiss et al. 2001). Interestingly, the study showed how temporal coordination mechanisms, defined as 'a process structure imposed to intervene and

direct the pattern, timing and content of communication', could mitigate the negative effect of avoidance and compromise behaviour (Montoya-Weiss et al. 2001, p.6). In this study compromise actually was shown to have a negative effect on team performance in distributed teams, which is surprising and exemplifies how researchers still lack a full understanding of the intricate dynamics of globally distributed work (Montoya-Weiss et al. 2001). Thus despite many individual research efforts providing useful insights into coordination in GSD teams, we still know little about the interaction between different coordination practices and when to apply them (Herbsleb 2007). In other words we still need to investigate cooperative work when teams have compatible processes across multiple sites to further our understanding of what practices are effective as discontinuities change.

To address this gap in the literature, one way is to investigate studies on coupling of work. Coupling of work has been of interest to CSCW researchers for more than a decade (Neale et al. 2004; Pinelle & Gutwin 2003; Herbsleb et al. 2000; Schmidt 2011). Coupling of work refers to the nature and degree of communication needed to solve a specific task (Olson & Olson 2000). In software development processes, the basic assumption has been that the degree of coupling of the work greatly impacts the need for coordination. If a task is loosely coupled it generally requires less articulation work and thus lessens the need for coordination strategies (Olson & Olson 2000). The characteristics of the task are thus closely related to the coordinative effort required. According to Olson and Olson (2000), loosely coupled work is characterized as simple standardized tasks with few interlinked dependencies, whereas tightly coupled work is ambiguous and is highly interdependent and thus very difficult to divide into smaller segments. Software products with many ambiguities and a high degree of dependency among the software components can be characterized as tightly coupled work.

Research proposes that loosely coupled tasks are more suitable for distributed work because they can more easily be divided into smaller sub-tasks (Hertzum and Pries-Heje 2011, Olson & Olson 2000). With fewer dependencies, tasks can be divided and reassembled again with only a minimum of coordination and interaction across the geographical distance in global software teams (Mockus & Weiss 2001, Grinter 2003). It has been argued that tightly coupled work is preferable when used in a collocated setting because collocated practitioners typically share a common ground and are better able to solve the ambiguities of tightly coupled work (Olson & Olson 2000). Contrary to this, other researchers have suggested that tightly coupled work is actually better suited for distributed work because it requires close connections and thus enables the practitioners to engage with each other on a mutually shared task (Bhat et al. 2006; Bjørn & Ngwenyama 2009). Thus this line of research points to the necessity for closely coupled work as an incentive for increasing interaction and forming partnerships among distributed team members.

The literature on coordination revealed that it becomes more difficult to manage dependencies in globally distributed teams. Managing dependencies requires both formal and informal ways of coordination, which is why researchers have suggested strategies such as minimizing links (Hertzum & Pries-Heje 2011) or segregation (Gerson 2008) to facilitate collaborative work in globally distributed teams. Clearly, there has been extensive research on supporting work by developing coordinative technologies but less focus on other aspects of coordination, for instance, through

informal talks and the structure of the work. Thus the challenges of coordination are closely linked to the nature of the task rather than technology. One interesting aspect was the fact that working in closely coupled teams on complex tasks in a distributed setting seems counterintuitive according to existing literature. The general assumption seems to be that the decoupling of components also allows a decoupling of tasks (Grinter 2003; Olson & Olson 2000). Recent research has come to question this assumption (Herbsleb 2007). However, decoupling of tasks does not necessarily facilitate conditions for global collaborative practices. Tightly coupled work may require more effort but it also enables closer connections that can work towards more productive collaboration and aid the continuous adjustment to recurrent discontinuities. Thus it becomes essential to understand the nature of the work and how the division of tasks impacts on the collaboration. Consequently, the role of coordination still requires more research to answer questions like: How does coupling of work in globally distributed settings enable the practitioners to adjust to existing discontinuities? What do closely coupled work practices entail? What are the enabling conditions for different degrees of coupling in globally distributed work settings? In summary we need to get a better understanding of the enabling conditions for making closely coupled work practices be effective in global software development teams. The core of these questions will be addressed in Paper 4.

3.5. Establishing a Mutual Understanding across Different Social Worlds

People working in geographically distributed settings have little opportunity to meet casually and informal discussions are thus limited. These are important for establishing a shared common ground. Common ground (Olson & Olson 2000) refers to the knowledge that people have in common and of which they are mutually aware. When IT developers work across geographical distances in cross-national teams, it becomes increasingly difficult to establish a mutual understanding. Typically, offshoring companies have their base of origin in Western countries (US and Europe) and have thus developed a client base located in these countries and developed expertise within Western organizations or municipalities (Aspray 2006; Šmite et al. 2010). GlobalSoft is an example of such a setup, where the majority of the clients were Danish municipalities. Consequently, domain knowledge was paramount in order to be able to develop software systems in that particular setting. The requirement specification is the primary artefact for communicating the domain specific information but it is also very challenging to manage in global software development (Damian & Zowghi 2003; Prikładnicki et al. 2003). When working in global team, the domain knowledge in the requirement specification requires a shared context to be interpreted correctly by the involved practitioners (Liang et al. 2009). The requirement specification in our case was constructed as a collection of documents in close collaboration with the client and it was used for defining and assigning tasks to members of the team. The client presented a vision of the final product and the Danish employees engaged in a discussion with the client on how to achieve this vision. Thus the information embedded in the document is based on negotiations with the client regarding the final product (Damian 2007).

Establishing a mutually shared understanding of the task at hand, is closely related to the culture in the sense that cultural differences are a major reason for lacking a common ground (Olson & Olson 2000). Consequently, understanding how

practitioners manage the concept of culture in global software development is linked to the challenges that the practitioners experience with sharing knowledge and creating common ground (Cramton & Hinds 2007; Krishna et al. 2004; King & Torkzadeh 2008). Researchers of global software development have employed different approaches to the 'cultural' phenomenon and much of the discussion evolves around how the concept of culture should be used analytically. The concept of culture is a contested issue and some researchers imply that culture is a social construction based on values, belief and norms (Søderberg & Holden 2002), whereas others describe culture as the national characteristics of a person (Hofstede et al. 2010). Management and business inspired research studies of culture often emphasize the impact of cultural differences in terms of national or organizational identities and propose ways to handle these differences (Y. Kim & S.-Y. Kim 2010; Keil et al. 2000). This type of studies often refers to Hofstede's notion of different dimensions of national culture for understanding disparities between, for example, Western and Asian countries. (Kale 1991, Gudykunst et al. 1996, Kim and Kim 2010). Hofstede presents five categories based on large quantitative interview studies, which are power, distance, individualism, masculinity, uncertainty avoidance, and long-term orientation. These categories can be applied to define national characteristics and have been widely used by companies to understand cultural misunderstandings.

However, Hofstede's work has also received much critique especially that these categories are too static and that this cultural approach becomes deterministic in its way of explaining cultural phenomena. Instead researchers have suggested that culture should be investigated as a social construction between people (Søderberg and Holden 2002). This approach allows researchers to apply a much more flexible understanding of culture as a construct that is susceptible to change over time. An individually based understanding of culture based on values, norms, and beliefs allows for a deeper understanding of how cultural distance impacts on distributed teams. Several global software development studies have moved away from national stereotypes and propose instead a much more nuanced understanding of culture. Alternatively, the concept of culture should be understood as 'a shared web of meanings that shapes roles and interpretations, and is dynamically (re)negotiated by the actors in the course of their daily work' (Boden et al. 2012). This interpretation of culture suggests a more flexible understanding of culture that is negotiated between the practitioners. This approach allows researchers to engage better in cultural studies and grasp the many ways in which culture has been found to impact on the collaborative work in cross-national teams.

Culture defined as a set of values, beliefs, or norms (Søderberg and Holden 2002) also implies that these values are linking the society in which the participants are situated – i.e. that aspects of the society shape individuals' interpretations of the social context. 'Context' can be understood in its broadest sense to include geographical location, organizational structures, and workplace setting (Ngwenyama & Bjørn 2007). It is similar for all three conceptual structures that they impact on and affect each other over time. When people from different social contexts meet, their interaction is likely to spark the formation of new assumptions, beliefs, and values that is the building blocks for a shared context. However, one of the challenges for building a shared context is the emergence of fault lines within groups (Cramton & Hinds 2005). Fault lines are characterized as the alignment of demographic attributes such as gender, age, cultural background, or professional expertise, which increase the

risk of sub-group dynamics in distributed teams. Sub-groups have the potential to create ethnocentric views of the other team members especially when distributed team members rarely meet and thus they hinder the establishment of a shared context. Ethnocentric views can impact negatively on the collaborative work by establishing a sense of ‘us and them’ in geographically distributed teams (Cramton & Hinds 2005). Negative sub-group dynamics within a global software development project can be damaging for the collaboration and result in conflicts as other studies of global software development projects have shown (see for example Marrewijk 2010, or Metiu 2006). These studies found that strong internal groupings, based on geographical locations, hindered the collaborative work or even caused the collaboration to end permanently. Negative sub-group dynamics can be mitigated by focusing on creating strong connections across distributed groups that establish cross-cutting fault lines across the groups located in the different locations (Cramton & Hinds 2005). The positive aspects of sub-groups are specifically achieved when fault lines are established between distributed team members that potentially can create cross-cultural learning (Cramton & Hinds 2005).

While the concept of culture is still widely contested (McSweeney 2002), there is still a profound need to investigate the notion of culture and how it impacts on the collaborative practices in global software development teams. Research has found that cultural differences can result in communication breakdowns (Krishna et al. 2004), the formation of sub-groups (Cramton & Hinds 2005) and they have even been reported to be used strategically in global software teams to maintain control of certain tasks and responsibilities (Ybema & Byun 2009; van Marrewijk 2010). For example, one study reported that team members in a global software development project used cultural constructions as a leverage to gain access to favourable tasks or roles in the project (van Marrewijk 2010). Marrewijk reports that Indian developers would point to the characteristics of Indian culture as harmonious, which in turn would make Indian developers more eligible for handling client relations. Likewise, the Dutch developers would point to their own cultural heritage to legitimise control over the organization of tasks (2010). Other studies of culture in global software development focus on the role of culture for knowledge sharing (Boden et al. 2012; Boden & Avram 2009; Kotlarsky & Oshri 2005). Yet other studies have focused on ways to mitigate cultural challenges, such as cultural learning (Krishna et al. 2004) or having boundary spanners in the software teams to ‘bridge’ the cultural gap (Avram et al. 2009). These changes in work practices seem to work well in certain cross-cultural setups.

However, recent research has begun to investigate culture from the perspective of power and post-colonial perspectives (Mahadevan 2011; Ravishankar et al. 2013), which enables a new discourse in global software development research of cultural differences. These examples suggest that culture is at play on a deeper level and is used to realize underlying motives among the team members. These studies imply an ‘unboxing’ of the cultural explanations based on an understanding of culture as a renegotiable construction that individuals apply in certain situations and these constructions shape how the practitioners experience the global collaborative work. Thus this is a step away from the static national presentations of culture that Hofstede represents but they are also different from studies using cultural explanations to explain breakdowns in the collaborative work. The common denominator for these studies is largely to investigate the challenges that occur in geographically distributed

teams due to different cultural backgrounds and thus placing culture at the fore of their studies. The findings of recent research suggest that before applying the concept of culture as an explanation, researchers have to be careful because underlying motives and negotiation of power have a great impact on the collaboration in global software teams (Metiu 2006).

The research literature on culture in global software development is still contested as people still discuss how to define and apply the concept of culture. In practice culture is being used to describe many different phenomena in global software teams and as researchers of global software teams, we need to ask how cultural practices are enacted in global teams. It is essential to understand how and when cultural differences create challenges for the collaborative work in order to be able to support better global work practices in software development teams. Thus these questions are being addressed in Paper No. 2. Moreover, there is a need to identify the link between the notion of culture and emergent social worlds in global software teams. In other words we need to investigate how social worlds become salient in the everyday practices and identify the practices of the culture which is being enacted in global software teams – questions that are being investigated in Paper No. 1.

4. SUMMARY OF THE PAPERS

This chapter is a brief introduction to the main part of this dissertation. The four selected papers will each be presented with a short summary. The full versions of the papers can be found in the second part of this dissertation.

4.1. Paper No. 1: Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds

In this paper the authors suggest that one way to capture the national perspective on culture without submitting to the categories of national culture is to think in terms of social worlds. The paper investigates cultural complexities as social worlds and asks the following research question:

How can we identify situations where the differences in social worlds between geographically distributed developers become salient in their everyday interactions?

In investigating the empirical material, we found several examples where concepts essential for the design of the IT system were not part of the social world of the remote site. For example, the concept of food and health inspectors that is a commonly used practice in Denmark, was entirely absent in the Philippines. We found that when knowledge, relevant for the interpretation of the system requirements, was localized in the social world of one location but not in the other, this could lead to communication breakdowns. In these examples, characterized as a high diversity situation, identifying the issue and resolving communication breakdowns became a process of explaining, negotiating, and creating a shared meaning context. In high diversity situations, developers working across sites might already be aware of the risk of misunderstandings and thus they will use considerable resources on translating domain specific knowledge. However, this analysis also revealed a different type of situation with less diversity. These situations emerged when a specific concept (such as pension systems or social security numbers) is shared across locations but has different meanings due to different social worlds. Communication breakdowns in these situations were caused by diversity in the meaning of shared concepts, which are more likely to happen in the later stages of a project because participants may perceive a ‘false’ sense of common ground, making the lack of shared understanding harder to identify.

By analyzing both interviews and observations, we identified two types of situations where social worlds become salient in the everyday interactions between developers working at different geographical locations: 1) The divergence of concept and meaning and 2) the convergence of concept but divergence of meaning. We propose that situations with divergence of both concept and meaning across social worlds are quite evident, so practitioners are aware of the differences. It is relatively easy to identify instances of divergence in concept because the lack of a shared vocabulary is obvious. In cases of convergence in concept but divergence in meaning, detecting a lack of common ground is difficult and, in some situations, even impossible. In many cases, the lack of common ground will not appear until technical decisions based on the false assumptions become manifested in the IT system. We argue that these situations are grounded in social worlds and pose a challenge to work practices in the form of miscommunications and misinterpretations of shared tasks. By identifying

and describing these situations, we can better understand how and why communication breakdowns occur in intercultural collaborative work practices.

4.2. Paper No. 2: The Rhetoric of Culture as an Act of Closure in a Cross-national Software Development Department

This paper investigates the rhetoric use of culture as an informal act of exclusion in cross-national software development teams. As a starting point for this study, we found that many employees in the Danish office questioned the rationale behind offshoring despite the rationale of improved competitiveness offered by the top management. The competitive edge derived from employing Filipino workers was a claim that remained unproven, as the company had not conducted a comparative study of the cost difference between work done at the primary location and work done offshore. Danish employees remained skeptical towards offshoring, and tried to maintain tasks and decision-making power locally through informal acts of closure. Recent research proposes that more research is needed regarding ‘how the actions of group members exclude others [even] when their official organization is committed to the cooperation’ (Metiu 2006). We therefore ask:

How are informal patterns of closure enacted in cross-national collaboration?

We show how higher status employees sought to protect their tasks and roles by invoking ‘culture’ as the root of collaboration problems. By status we are referring to the relative decision power over tasks. The Danish office had a history of being solely responsible for the tasks and found the transition towards global collaboration threatening and difficult. In this study we observed that questions of culture revolved not around cultural differences but rather issues of power and influence. We found that ‘culture’ was invoked by employees in the Danish office to explain failed collaborations but not by employees in the Philippines. Employees located in the Philippine department gave other explanations for problems, and actively resisted cultural explanations. For example, when communications broke down, employees in the Danish office would vaguely explain this by cultural differences whereas employees in the Philippine office would point to problems with the existing communication practices.

The data material revealed that in construing national culture as a stable, persistent condition, options for negotiation and discussion, which might have brought the Danish and Philippine employees into a state of more equitable relations, were foreclosed, and static relations tended to persist. We argue that employees in the Danish office used ‘rhetoric of culture’ as an act of closure to preserve the existing ways of working. Closure affected media choices and patterns of everyday communications. In particular, we observed how modes of asynchronous mediated communications enabled acts of closure through the rhetoric of culture whereas video conferences seemed to promote collaboration through the necessity of commitment, and possibly through the face-to-face nature of video conferencing. An aim of our work is to point to the rhetorical use of the word ‘culture’ as an act of closure in cross-national distributed work, something not yet reported in the literature as far as we know. By applying a sociological perspective to IT offshore studies we also contribute with much needed insights into the cultural and informal management mechanisms in cross-national teams.

4.3. Paper No. 3: Demonstrating Commitment in Practice: Tracing & Gaining Attention

In this paper we explore the ways in which individual practices demonstrate commitment as part of a collaborative engagement, and how these practices become part of the collaborative work arrangement. The majority of prior research on commitment within CSCW takes the view that commitment is a discursive construct and, within this approach, the design for commitment tends to produce certain strategies based upon the categorization of declarations of commitment. We wanted to take a different approach and investigate what it would mean to think about commitment as a practice, rather than as a discursive construct.

Investigating the data material from a global software development project, we observed that practitioners are tracking attention using two types of traces, namely temporal and artefactual. These traces were attempts to establish collective attention concerning the task, the process, and the people involved, thus providing important information about remote colleagues. Moreover, practices that demonstrate commitment are concentrated around tracing remote colleagues' attention through temporal and artefactual clues captured by technology mediation. We found that the practices of tracking traces were not only about knowing the availabilities of others but also concerned the ways in which individuals were able to demonstrate commitment and how these practices of commitment became part of the collaborative work arrangement. Demonstrations of commitment are founded on attempts to establish collective attention among participants concerning the task, the process, and the people involved. To comprehend commitment as a feature of collaborative work fully rather than as a declarative state of mind, we needed to move from the perspective of *declaring commitment* towards the perspective of *demonstrating commitment*. The differences between these two perspectives might appear simple but in fact this rephrasing transforms the basic nature of what commitment is and how we could investigate it as part of collaborative practice. In the perspective of demonstrating commitment, it is clear that commitment is not only a mental or discursive construct but also part of everyday practice.

This paper contributes with two main points. Firstly, we identified two different strategies, namely the practices of enacting temporal and artefactual traces within the technologies by which the geographically distributed IT developers manage to demonstrate commitment. Secondly, we propose a conceptual understanding of commitment as a practice instead of a discursive construct. Thus demonstrations of commitment are an important feature of collaboration, which complements and further develops the basic foundations for how we can understand collaborative practices in CSCW research.

4.4. Paper No. 4: Why Closely Coupled Work Matters in Global Software Development

This study investigates why IT developers in a Danish-Philippine collaboration chose to engage in more closely coupled work as the project progressed. As a starting point for this study, we observed how the IT developers characterized the collaboration as the best to date. This was quite surprising given the fact that the project had failed on

traditional parameters, such as economy and time. Moreover, we had observed a gradual shift towards closely coupled work practices and thus the research question asked:

Why did the IT developers choose to engage in more closely coupled work?

The data material revealed a gradual transition in the work practices over a period of three years. Over time the IT developers facilitated more closely coupled work, which became essential to solve the task. This transition was not carefully orchestrated by the management group but rather carried out as a non-sequential process that involved a range of different employees in the company.

This paper points to three key factors of how the collaboration improved during the transformation from loosely to closely coupled work. Firstly, closely coupled work practices established many connections across the collaboration ensuring knowledge exchange and improving coordination, for instance, by establishing daily test meetings, collocating people, and formalizing communication procedures between developers.

Secondly, closely coupled work practices diminished the formation of sub-groups locally and established new fault lines across the geographical distance by removing the organizational constraints of the fixed price model and intensifying travels between the two locations. By collocating Danish employees and increasing the frequency of meetings between distributed members, the IT developers managed to moderate patterns of ‘us and them’ both locally and remotely in the company.

Finally, closely coupled work created connections across organizational hierarchies by introducing daily ‘board’ meetings and new project leader meetings. These meetings increased awareness of the complexities of the project, made people accountable and enabled shared responsibility for the project outcome. Moreover, the connections across organizational hierarchies allowed information to travel seamlessly between layers in the organization and consequently the practitioners could better anticipate problems and act accordingly, for instance by allocating more resources to the project.

In conclusion, closely coupled work helped the IT developers in solving the task despite being very challenging and suffering from delays and financial losses. Not only did they solve the task, they also experienced that the collaborative work was the most successful to date. Thus transforming work practices from loosely coupled to closely coupled practices became a valuable learning process for executing global software development in the company.

5. DISCUSSION

Previous research is quite clear on one aspect of global software development, namely that it is not easy to make it work (Herbsleb et al. 2005; Noll et al. 2010; Jensen & Bjørn 2012; Matthiessen et al. 2014). It requires a learning process to handle the intertwined mix of dependencies that includes both human and technological factors (Boden et al. 2012; Gregory 2010). While sophisticated tools and technologies for distributed work have been widely available for at least a decade, they do not seem to solve the core issues of global collaboration. Instead answers to the research question will be found by focusing on what can be described broadly as the ‘human factors’. When looking into the empirical observations and asking how to coordinate the work to facilitate frequent changes in global software development. One important aspect detected was the gradual shift towards more closely coupled work practices over the course of the three years. This shift can be divided into three main stages.

The first stage described the transition from an outsourcing to an offshoring collaboration that took place in 2009. The acquisition of the Philippine office facilitated a merger of the Danish and the Philippine offices and removed the discontinuity of working across different companies. However, discrepancies in social worlds made communications across locations difficult and many practitioners in GlobalSoft experienced that the teams were working as two individual teams with little success. Communication breakdowns occurred often when differences in social worlds became salient. This was disruptive for the collaborative practices. We saw this exemplified as the divergence in the social worlds (Paper No. 1).

The second stage involved the changes in the overall organizational structures, such as the changes in the contractual setup as well as the travel policies that took place in the beginning of 2011. Here it was apparent that these prior organizational practices concerning travel and the contractual setup had maintained some of the discontinuities from the outsourcing days and a feeling of ‘us and them’ remained between the Danish and the Philippine offices. Thus the organizational structures, which were disruptive for the collaborative work experience, were removed. This was seen as a critical change for many of the IT developers in the DAFIC project (Paper No. 4). The main advantage of getting access to more travelling was that project members were able to meet each other face-to-face and could build social connections and create common ground (Olson & Olson 2000), and commitment (Nardi 2005). The organizational context for working in remote teams is important in terms of whether it enables or constraints the collaboration (Bjørn & Ngwenyama 2009). Thus this change was important for improving the collaboration across the existing sub-groups that had emerged across the two locations in the DAFIC project. Nevertheless, it turned out that the Danish developers were not interested in traveling that much, and more importantly team members in both offices regularly left the project and potentially eliminating the already established connections as seen in previous studies (Boden et al. 2009). Thus changing organizational practices such as the contractual setup and implementing liberal travelling policies were not sufficient to make the collaboration work well as exemplified in the rhetorical use of cultural differences (Paper No. 2). Consequently, it became evident in this empirical case that the organizational practices did not have the anticipated outcome even though they were perceived as being critical for making the collaboration work.

This is where the third and final stage became critical for the global collaborative work in the DAFIC project. The third stage describes a gradual push towards more closely coupled work in the DAFIC project beginning from 2012 and increasing towards the end of the project in 2014. The data material showed how the practitioners began to have contact on a regular basis in terms of both being physically present for extended periods of time at the other location or by scheduled video conference meetings. Testers who initially only had sporadic contact by email and chat, began to have daily video conferences towards the end of the project (Paper No. 3). On the Danish side they implemented daily ‘board’ meetings, including both managers and IT developers and they also moved people so everybody on the Danish side were collocated in the same building at the end of the project. In the Philippines they began to work in groups instead of individually on a project. Lastly, regular project leader meetings were established. They were attended not only by the project leaders but also the IT developers and managers from both offices (Paper No. 4).

5.1. Persistent Discontinuities in Global Software Development Teams

It is already well documented in GSD research that discontinuities disrupt everyday work patterns and make it more difficult to collaborate (Noll et al. 2010). For example, geographical distance is a typical discontinuity in collaborative work because it alters the patterns of cooperation compared to collocated collaboration. Geographically distributed team members cannot meet as often as collocated team members and will have to use technology mediated means of communication. Similarly, discontinuities, such as time difference, cultural diversity and variance in expertise, may create limits for the team members’ collaboration making it challenging to make the collaboration work. In the empirical data investigated in this dissertation, the IT developers collaborated across a range of discontinuities, namely geographical distance, organizational difference, cultural diversity, as well as changing team composition. Continuities emerge when practitioners familiarize themselves and thus reach a state where discontinuities no longer have any impact on the virtual collaboration because the practitioners have established shared norms and expectations based on continuous interaction. However, the data material showed that frequent changes were an integral part of the everyday work that arguably made it more difficult to maintain shared norms and expectations in the project. In order to explain this a little further, we need to investigate the persistent discontinuities in global software teams by looking at the frequent changes observed in the DAFIC project.

Observations in the project revealed that the DAFIC project constantly experienced changes on both the organizational level and in the team composition. Some of these changes were extraordinary while other presented themselves as normal natural problems in the global software teams. As an example of the latter, the data material revealed that team members left the project on a regular basis. Some team members found better job opportunities or got fired, while others left because of personal issues such as stress or maternity leave. More specifically, the project leader was replaced three times in Denmark whereas the Philippine office had two official project leaders and a stand-in. The DAFIC project had three different chief developers in the Philippine office and the Danish office replaced the test manager as well as a number of experienced developers. The frequent turnover made it very difficult to maintain shared expectations and consequently establish continuity (Watson-Manheim et al.

2002). Moreover, internal processes and company policies were also evolving and the client had their representatives replaced several times during the project. In addition to these frequent – yet normal changes - extraordinary changes also occurred during the course of the project. These extraordinary changes included moving to new office locations that consequently relocated and divided groups of project members (Paper No. 4). During the three years in GlobalSoft, the IT developers changed office location three times and had to adjust accordingly. Sometimes this affected access to servers whereas at other times it separated the IT developers from the back office team. Changes in the team composition were a regular occurrence throughout the project and therefore emerged as a persistent discontinuity in collaborative work.

The most critical changes for the collaboration occurred when people left the project entirely. The data material revealed – perhaps not surprisingly – that loosely coupled work patterns concentrated the expertise and knowledge of specific components on fewer individuals. The DAFIC project consisted of more than 5000 pages of documentation, which was an insurmountable amount of information especially for new people. Consequently, when team members left the project, knowledge as well as social relations accumulated over time were disrupted or even completely disregarded, creating discontinuities in the work. The data material from this study suggests that discontinuities may remain as a constant factor of globally distributed software development. The idea that the team will stabilize in terms of people, locations and internal processes, did not seem to be viable, since such changes seemed to be normal natural problems in a global software development project.

5.1. Increasing Dependencies with Closely Coupled Work

The empirical observations reported upon in this dissertation, question the notion that discontinuities will fade into the background and become gradually unnoticed. The published papers included in this dissertation show that discontinuities never seemed to disappear – but continuously emerged in new ways. Therefore, it would be futile to aim at attempting to transform discontinuities into continuities. Instead, we need to change the perspective in order to figure out how to improve collaboration in GSD work despite discontinuities such as frequent changes in the team composition. The data material showed that it was critical to establish some stability in a constantly changing project environment. However, this stability did not emerge as shared norms and expectations, because people were leaving the project and were replaced by others. Instead the data material indicated that discontinuities seemed to be a constant factor in the collaborative work. Thus researchers will have to discard the idea of the ‘ideal’ collaborative setting, because it will most likely never occur and only for short periods of time in global software development teams. Instead IT developers will have to be able to adjust and adapt to emergent changes in the collaboration (Boden et al. 2012) and one approach will be to establish stability through coordination of work in terms of closely coupled work practices, despite the inherent change of membership patterns in cooperative ensembles (Schmidt & Bannon 1992).

Investigating the path of closely coupled work as a vehicle to support global software teams, the data showed that it enabled the practitioners to adjust when people left and also become more resistant to the constant discontinuity of working with new people (Paper No. 4). Moreover, we also observed how closely coupled work practices enabled the IT developers to establish a collective attention and demonstrate

commitment in the project better (Paper No. 3). It requires an increase in dependencies *across* locations to apply closely coupled work practices in a global software development team. Instead of segregating tasks (Gerson 2008) or decomposing software (Grinter 2003) as a means of coordinating the work, the findings in this dissertation support the idea of increasing dependencies across locations (Bjørn & Ngwenyama 2009). Such an idea may seem counterintuitive (Hertzum & Pries-Heje 2011) because an increase in dependencies will necessarily result in more articulation work in order to coordinate the work. However, coordination mechanisms, such as project plans, cannot reliably support the articulation work alone (Whittaker & Schwarz 1999) especially when frequent changes are part of the normal natural problems in global software teams. Moreover, articulation work requires continuous interaction between team members, especially in small to medium-sized teams. This cannot be solely handled by technical solutions but also requires informal patterns of communication (Boden et al. 2007).

Increasing dependencies complicate some aspects of the work but also enable the formation of more connections across the team and create incentives for interaction. In globally distributed teams working in domain knowledge intensive fields, such as developing software to a Danish context, increased interaction became important for productive collaboration. Domain knowledge was critical for the Danish-Philippine collaboration (see Paper No. 1) and insisting on closely coupled work practices can become a coordinative measure that migrates instead of separating the globally distributed team. Efforts that integrate the globally distributed teams were important to alleviate the initial feeling of 'us and them' in the Danish-Philippine collaboration and could potentially diminish the rhetoric use of culture observed in Paper No. 2. While closely coupled work alleviated negative sub-group dynamics in the project by establishing more connections there may be more to gain by identifying the characteristics of closely coupled work.

5.2. Characteristics of Closely Coupled Work

As previously described, the Philippine company became an integral part of GlobalSoft during the first stage thus removing the discontinuity of working across different companies. The data material revealed that despite officially being a single company, the two offices still worked largely as independent actors linked together solely by a financial obligation (Paper No. 2). Notwithstanding the changes in the organizational structures and the on-going collaboration between Danish and Philippine developers for more than three years prior to the acquisition of the Philippine office, many of the projects still suffered from divergence in either concept, meaning, or both simultaneously (Paper No. 1). Thus the challenge of developing to a specific domain still prevailed in the projects causing misunderstandings and miscommunications. During the second stage, the contractual setup was changed allowing for the integration of the financial responsibility in projects between the Danish and Philippine offices. Travel policies also became less restrictive to enable more face-to-face contacts in the distributed collaboration. Even so collaboration was still struggling during the second stage. For instance, the empirical data showed that the employees in the Danish office used national culture as an explanation for miscommunications, which foreclosed negotiations and narrowed the scope for solutions (Paper No. 2). At this point in time projects shared a financial responsibility and people were able to meet more regularly. However, tasks were still

largely divided between practitioners with few dependencies across locations. The shared financial responsibility and organizational structures were still important conditions enabling the progress towards more closely coupled work practices in the DAFIC project.

Stage three illustrates a gradual shift towards working more closely together (Paper No. 4). While it may not be feasible to point to a single reason why this shift occurred in a complex project spanning more than three years, it was clear that the developers on both sides experienced a growing dependency on each other in order to solve the task. Dependency increased as the project became more delayed and financially pressured making it less and less feasible simply to cancel the collaboration and develop everything at a single site, mainly because specific knowledge and expertise on specific software components had accumulated on both sides. After the initial strategy of dividing the software product into four components to minimize the dependencies across locations, this resulted in a situation where the IT developers were highly dependent on each other (Paper No. 4). Mutual dependency in the global collaboration required the IT developers to find ways of overcoming the challenges, such as working across different social worlds (Paper No. 1), and to diminish the excluding behaviour observed in Paper No. 2. As a result the IT developers engaged in more closely coupled work practices both horizontally across geographical locations in the project but also vertically across the organizational hierarchy (Paper No. 4). As an example of closely coupled work practices, the IT developers initiated daily meetings involving developers as well as project leaders and managers established many connections across both locations in the project and kept everybody informed of the general progress of the project (Paper No. 4).

Locally, people were relocated from different offices to work together in the same office and responsibilities for different components were shared among several people instead of just one person. The testers began to collaborate more closely by having daily meetings followed up with continuous messages throughout the day and summarizing everything in status emails towards the end of the day. Such practices enabled the testers on both sides to track traces of attention and thus establish a collective attention and demonstrate commitment to complete the task (Paper No. 3). A key feature of the closely coupled work practices was going from sporadic interactions to constant and daily interactions across both locations, between not just a few people but many different people in the project. Becoming mutually dependent on the task and being able to trace other people's attention exemplify the gradual shift towards more closely coupled work practices in the DAFIC project. In this particular case, the IT developers reached a point during the third stage where they described the cross-cultural collaboration as the most successful to date. The success was not in terms of financial gain or timely delivery of the project. Instead the collaboration served as an example of productive collaborative practices leading to constructive collaboration between the Danish and the Philippine offices.

5.3. Adjusting to Frequent Changes in the Collaborative Work

So how did the coordination of the work into closely coupled work practices enable the IT developers to make the collaboration work despite working in a disruptive working environment? Discontinuities may not transition into continuities in a setting such as a dynamic global software development team because people are not a stable

entity in globally distributed teams and building shared norms and expectations takes time. However, this is not to say that the effort to *adjust* will not prevail.

Coordinating the work by increasing dependencies across locations and among several team members, has the potential to create a stable structure in a work setting with many discontinuities where the practitioners are able to quickly adapt and adjust. With more connections across the collaboration, this enables the practitioners to develop, maintain and repair shared expectations faster and working closely together, making the project more resistant when people leave the project (Paper No. 4). Moreover, with multiple connections horizontally and vertically in the project, this enables the practitioners to respond accordingly and quickly catch up when people leave. A recent study describes distributed collaboration as a continuous learning process where the team members learn and adapt to discontinuities, such as cultural diversity (Boden et al. 2012). While there is *learning* involved in making the global collaboration work, it also is important to create conditions that allow learning and preserve the shared knowledge when people eventually leave the project.

In this dissertation the data material revealed that closely coupled work required shared financial responsibilities from all parties involved and enabling organizational structures such as access to travelling. Last but not least, practitioners must have a shared task with mutual dependencies. Such dependencies can be created at the beginning of the project when tasks are assigned or may develop over time as the project evolves as seen in the DAFIC project. The benefits of increasing dependencies in teams with the right conditions, are that it enables teams to be more resistant to frequent changes in the team composition, allow knowledge sharing through multiple connections and facilitate rapid adjustments due to connections across hierarchy. More connections across geographical locations alleviate sub-group dynamics and create a level of stability in work environment encumbered with discontinuities. The only caveat is that closely coupled work practices can seem very time-consuming and the learning process can be frustrating especially during the early stages of a project. It is an interactive, intensive approach requiring practitioners to use a range of communication tools, such as video conferences, instant messages and email while also occasionally being able to meet face-to-face. The data material suggests that closely coupled work practices are most useful for small to medium-sized teams operating in a field that requires specific domain knowledge and views the collaboration as a long-term investment. Larger teams with more standardized tasks may find this approach too time-consuming or costly compared to the relative gains. However, the findings from this study strongly advocate that small to medium-sized globally distributed teams greatly benefit from closely coupled work as a way of coordinating the work across the team instead of relying on dispersing and separating the work.

6. CONCLUSION

On behalf of the Danish project I have tried to advocate and argue that our Filipino colleagues lacked the necessary skills, but this project has made us all realize that we do not have the enough time to accomplish it alone and we have to trust that our Filipino colleagues can help us. Yes, it has been a costly affair in terms of the extra time it required, but they have been able to help us. This project has paid dearly for it, but I hope that this process will benefit us in the long term. We have learned that our Filipino colleagues are able to solve the tasks and we just have to learn how to include them in the right way. We may not be there yet, but it will come eventually. I believe so. (Danish Project leader for DAFIC)

This dissertation has been divided into several aspects concerning the frequent changes that occur as normal natural problems in global software development teams. Looking back at the research question, the aim was to investigate how IT developers coordinate the work to facilitate frequent changes in global software development projects. Investigating the data material from an ethnographic study of a Danish-Philippine software development project, the data showed that frequent changes in the team composition create persistent discontinuities that make it challenging to establish a shared context and mutual knowledge. Thus discontinuities did not reliably transition into continuities over time, because the establishment of shared norms and expectations was disrupted when IT developers frequently left the project. Despite these persistent changes the project members managed to create coordinative practices allowing them to experience the best collaboration to date. It was interesting to note that access to technology and tools rarely seemed to concern the practitioners as challenges were often grounded in the coordination of the work. Rarely if ever was lack of tools used as an explanation for breakdowns. Instead the project members had to find ways of coordinating the work to be able to act and adjust accordingly to the changes in locations, processes or people that occurred. The data material revealed a gradual shift towards more closely coupled work practices over the course of the three years in order to make the collaboration work.

Closely coupled work became a method for the practitioners in the DAFIC project to coordinate the work and reduce the complexity of discontinuities. The enabling conditions for this were first of all shared financial responsibility between the Danish and the Philippine offices. This was the first step towards more interdependence in the project. Secondly, the organizational structures should be geared to facilitate global collaboration with flexible travel policies to alleviate negative sub-group dynamics. Lastly, the organization of the task should emphasize mutual dependencies across locations to establish an incentive for more interactions between the project members.

The data revealed that changes in the organizational structures alone, such as the contractual setup and the travel policies, were not enough (Paper No. 1). It is critical to establish connections across geographical and organizational hierarchy. Closely coupled work created more connections across both the project and between managers and IT developers (Paper No. 4). The connections across alleviated the negative sub-group dynamics and the feeling of ‘us and them’ that had dominated previous projects (Paper No. 2). More dependency in the work increased the incentive for interaction

and practices made it easier for the IT developers to trace collective attention in the everyday work (Paper No. 3). Finally, having more connections made the team more resistant to the consequences of people leaving the project. In vastly complex software development projects, there is a risk that it is increasingly difficult to replace people. Sharing dependencies facilitated an easier learning process for new project members and was thus essential for completing the task.

In conclusion, closely coupled work practices created and maintained stability to enable the IT developers to adjust to the frequent changes in the global software development project. Consequently, this facilitated a solid base for collaboration despite working across various discontinuities. The enabling conditions for closely coupled work practices require a combination of shared financial responsibilities; organizational structures and mutual dependencies in the teams facilitate closely coupled work practices. In order to adjust to the frequent changes in global software development, it is important to structure the work in such a way that it enables stability not based on specific persons but specifically on closely coupled work practices.

While frequent changes occur as normal natural problems in dynamic global software development projects, causing disruption for the establishment of shared norms, expectations, and knowledge, working closely together enables new people quickly to become integrated into the team and to catch up. When more people are responsible for the task, there is less risk of losing essential knowledge when people leave the project.

Closely coupled work practices become critical for small to medium-sized teams operating in a field that requires specific domain knowledge and views the collaboration as a long-term investment to enable practitioners to handle and adapt to the inevitable constant discontinuities.

7. References

- Ackerman, M.S., Dachtera J., Pipek, V. & Wulf, V., 2013. Sharing Knowledge and Expertise: The CSCW View of Knowledge Management. *Journal of Computer Supported Cooperative Work*, 22(4-6), pp.531–573.
- Al-Ani, B. & Redmiles, D., 2009. In Strangers We Trust? Findings of an Empirical Study of Distributed Teams. *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering, IEEE Computer Society*, 2009, Washington, DC, USA. pp.121–130.
- Altheide, D. & Johnson, M., 1998. Reflections on Interpretive Adequacy in Qualitative Research. In Denzin, N. K. & Lincoln, Y. S., eds. *Collecting and Interpreting Qualitative Materials*, 4th edition, SAGE Publications, Inc.
- Anderson, R.J., 1994. Representations and Requirements: The Value of Ethnography in System Design. *Human-Computer Interaction*, 9(3), pp.151–182.
- Aspray, W., Mayadas, F. & Vardi, M. Y, eds., 2006. *Globalization and Offshoring of Software*, ACM Press.
- Avram, G., Bannon, L., Bowers, J., Sheehan, A. & Sullivan, D.K., 2009. Bridging, Patching and Keeping the Work Flowing: Defect Resolution in Distributed Software Development. *Computer Supported Cooperative Work*, 18(5-6), pp.477–507.
- Bentley, R., Hughes, J.A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D. & Sommerville, I., 1992. Ethnographically - Informed Systems Design for Air Traffic Control. *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*, 1992, Toronto, Canada: ACM Press, pp.123–129.
- Bergvall-Kåreborn, B. & Howcroft, D., 2013. The Apple Business Model: Crowdsourcing Mobile Applications. *Accounting Forum*, 37(4), pp.280–289.
- Bhat, J.M., Gupta, M. & Murthy, S.N., 2006. Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing. *IEEE Software*, 23(5), pp.38–44.
- Bjørn, P. & Christensen, L.R., 2011. Relation Work: Creating Socio-Technical Connections in Global Engineering. *Proceedings of the 12th European Conference on Computer Supported Cooperative Work*, 2011, Aarhus, Denmark, pp.133–152.
- Bjørn, P. & Ngwenyama, O., 2009. Virtual Team Collaboration: Building Shared Meaning, Resolving Breakdowns and Creating Translucence. *Information Systems Journal*, 19(3), pp.227–253.
- Blomberg, J. & Giacomi, J., 1993. *Participatory Design: Principles and Practices* D. Schuler & A. Namioka, eds., Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.
- Blomberg, J. & Karasti, H., 2013. Reflections on 25 Years of Ethnography in CSCW. *Computer Supported Cooperative Work (CSCW)*, 22(4-6), pp.373–423.
- Boden, A. & Avram, G., 2009. Knowledge Management in Distributed Software Development Teams: Does Culture Matter? *International Conference on Global Software Engineering*. Limerick, Ireland, pp.18–27.
- Boden, A., Rosswog, F., Stevens, G., & Wulf, V., 2014. Articulation Spaces: Bridging the Gap Between Formal and Informal Coordination. *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 2014, Baltimore, Maryland, USA, pp.1120–1130.
- Boden, A., Avram, G., Bannon, L., & Wulf, V., 2012. Knowledge Sharing Practices and the Impact of

- Cultural Factors: Reflections on Two Case Studies of Offshoring in SME. *Journal of Software Evolution and Process*, 24(2), pp.139–152.
- 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*, Leipzig, Germany, pp.21–24.
- Boden, A., Nett, B. & Wulf, V., 2007. Coordination Practices in Distributed Software Development of Small Enterprises. *Proceedings of the Second IEEE International Conference on Global Software Engineering*, 2007, Munich, Germany, pp.235–246
- Boden, A., Nett, B. & Wulf, V., 2009. Trust and Social Capital: Revisiting an Offshoring Failure Story of a Small German Software Company. In I. Wagner et al., eds. *European Conference on Computer-Supported Cooperative Work*. Vienna, Austria: Springer, pp. 123–142.
- Brewer, J.D., 2000. *Ethnography*, United Kingdom, Open University Press.
- Bryman, A., 2004. *Social Research Methods*, United Kingdom, Oxford University Press.
- Carmel, E., 1999. *Global Software Teams*, USA, Prentice Hall.
- Carmel, E. & Tjia, P., 2005. *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*, United Kingdom: Cambridge University Press.
- Conchúir, E.Ó., Ågerfalk, P.J., Olsson, H.H. & Fitzgerald, B., 2009. Global Software Development: Where Are the Benefits? *Communications of the ACM*, 52(8), pp.127–131.
- Cramton, C.D. & Hinds, P., 2007. Intercultural Interaction in Distributed Teams: Salience of and Adaptations to Cultural Differences. *Proceedings of Academy of Management*, August 2007, (Meeting Abstract Supplement), pp.1-6.
- Cramton, C.D. & Hinds, P., 2005. Subgroup Dynamics Internationally Distributed Teams: Ethnocentrism or Cross-National Learning? *Research in Organizational Behavior*, 26, pp.233–265.
- Damian, D., 2007. Stakeholders in Global Requirements Engineering: Lessons Learned from Practice. *IEEE Software*, 24(2), pp.21–27.
- Damian, D.E. & Zowghi, D., 2003. RE Challenges in Multi-Site Software Development Organisations. *Requirements Engineering*, 8(3), pp.149–160.
- de Souza, C.R.B. & Redmiles, D.F., 2011. The Awareness Network, To Whom Should I Display My Actions? And, Whose Actions Should I Monitor? *IEEE Transactions on Software Engineering*, 37(3), pp.325–340.
- de Souza, C.R.B. Quirk, S., Trainer, E., & Redmiles, D.F., 2007. Supporting Collaborative Software Development Through the Visualization of Socio-Technical Dependencies. *Proceedings of the International ACM Conference on Supporting Group Work*, 2007, Sanibel Island, USA, pp. 147–156.
- Dekel, U. & Herbsleb, J.D., 2008. Pushing Relevant Artifact Annotations in Collaborative Software Development. *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, San Diego, CA, USA, pp.1-4.
- Ebert, C. & De Neve, P., 2001. Surviving Global Software Development. *IEEE Software*, 18(2), pp.62–69.
- Esbensen, M. & Bjørn, P., 2014. Routine and Standardization in Global Software Development. *Proceedings of the International ACM Conference on Supporting Group Work*, 2014, Sanibel

- Island, USA, pp.12–23.
- Espinosa, J.A., Slaughter, S., Kraut, R., & Herbsleb, J., 2007. Team Knowledge and Coordination in Geographically Distributed Software Development. *Journal of Management Information Systems*, 24(1), pp.135–169.
- Espinosa, J.A., Cummings, J.N., Wilson, J.M., & Pearce, B.M., 2003. Team Boundary Issues Across Multiple Global Firms. *Journal of Management Information Systems*, 19(4), pp.157–190.
- Feyerabend, P., 1993. *Against Method*, 3rd edition, London, Verso.
- Fitzpatrick, G., Marshall, P., & Phillips, A., 2006. CVS Integration with Notification and Chat Lightweight Software Team Collaboration. *Proceedings of the 2006 20th Anniversary Conference on Computer-Supported Cooperative Work*, Banff, Alberta, Canada, pp.49–58.
- Gerson, E.M., 2008. Reach, bracket, and the Limits of Rationalized Coordination: Some challenges for CSCW. Ackerman, M.S., Halverson, C.A., Erickson, T., & Kellogg, W.A., eds. In *Resources, Co-Evolution and Artifacts: Theory in CSCW*, Springer, pp.193–220.
- Gregory, R., 2010. Review of the Offshoring Literature: The role of Cross-Cultural Differences and Management Practices. *Proceedings of the European Conference on Information Systems*, 2010, Pretoria, South Africa, paper 147.
- Grinter, R.E., 2003. Recomposition: Coordinating a Web of Software Dependencies. *Journal of Computer Supported Cooperative Work*, 12, pp.297–327.
- Grinter, R.E., Herbsleb, J.D. & Perry, D.E., 1999. The Geography of Coordination: Dealing with Distance in R&D Work. *Proceedings of the international ACM SIGGROUP conference on Supporting group work*, 1999, Phoenix, Arizona, USA, pp.306–315.
- Gross, T., 2013. Supporting Effortless Coordination: 25 Years of Awareness Research. *Journal of Computer Supported Cooperative Work*, 22(4-6), pp.425–474.
- Gutwin C., Penner, R., & Schneider, K., 2004. Group Awareness in Distributed Software Development. *Proceedings of the 2004 ACM Conference on Computer-Supported Cooperative Work*, Chicago, Illinois, USA, pp.72–81.
- Handel, M. & Herbsleb, J.D., 2002. What is chat doing in the workplace? *Proceedings of the 2002 ACM Conference on Computer-Supported Cooperative Work*, New Orleans, Louisiana, USA, pp.1–10.
- Harper, R.H.R., 2000. The Organisation in Ethnography. *Computer-Supported Cooperative Work*, 9(2), pp.239–264.
- Heath, C. & Luff, P., 1992. Collaboration and Control: Crisis Management and Multimedia Technology in London Underground Line Control Rooms. *Journal of Computer-Supported Cooperative Work*, 1(1), pp.24–48.
- Herbsleb, J.D., 2007. Global Software Engineering: The Future of Socio-Technical Coordination. *Future of Software Engineering*. IEEE, pp.188–198.
- Herbsleb, J.D., Mockus, A., Finholt, T.A. & Grinter, R.E., 2001. An Empirical Study of Global Software Development: Distance and Speed. *Proceedings of the International Conference on Software Engineering*, IEEE Computer Society, 2001, Washington, DC, USA, pp.81–90.
- Herbsleb, J.D., Mockus, A., Finholt, T.A. & Grinter, R.E., 2000. Distance, Dependencies and Delay in a Global Collaboration. *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, 2000, Philadelphia, Pennsylvania, pp.319–328.

- Herbsleb, J.D., Paulish, D.J. & Bass, M., 2005. Global Software Development at Siemens: Experience From Nine Projects. *Proceedings of the International Conference on Software Engineering*, 2005, St. Louis, MO, USA, pp.524–533.
- Hertzum, M. & Pries-Heje, J., 2011. Is Minimizing Interaction a Solution to Cultural and Maturity Inequality in Offshore Outsourcing? In Hertzum, M. & Jørgensen, C., eds. *Balancing Sourcing and Innovation Systems Development*. Trondheim, Norway: Tapir Academic Press, pp.77–97.
- Hofstede, G., Hofstede, G.J. & Minkov, M., 2010. *Cultures and Organizations: Software of the Mind*, 3rd edition, McGraw Hill Professional.
- Holmström, H., Conchúir, E.Ó., Agerfalk, P.J. & Fitzgerald, B., 2006a. Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance. *Proceedings of the IEEE International Conference on Global Software Engineering*, 2006, Washington, DC, USA, pp. 3–10.
- Holmström, H., Fitzgerald, B., Agerfalk, P.J. & Conchúir, E.Ó., 2006b. Agile Practices Reduce Distance in Global Software Development. *Information Systems Management*, 23(3), pp.7–18.
- Irani, L.C. & Silberman, M.S., 2013. Turkopticon: Interrupting Worker Invisibility in Amazon Mechanical Turk. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, Paris, France, pp.611–620.
- Jarvenpaa, S. & Leidner, D., 1999. Communication and Trust in Global Virtual Teams. *Organization Science*, 10(6), pp.791–815.
- Jensen, R.E. & Bjørn, P., 2012. Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds. *Proceedings of the 10th International Conference on the Design of Cooperative Systems*, 2012, Marseilles, France, pp.123–136.
- Keil, M., Tan, B., Wei, K.-K., Saarinen, T., Tuunainen, V. & Wassenaar, A., 2000. A Cross-Cultural Study on Escalation of Commitment Behavior in Software Projects. *MIS Quarterly*, 24(2), pp.299–325.
- Kim, Y. & Kim, S.-Y., 2010. The Influence of Cultural Values on Perceptions of Corporate Social Responsibility: Application of Hofstede's Dimensions to Korean Relations Practitioners. *Journal of Business Ethics*, 91(4), pp.485–500.
- King, W.R. & Torkzadeh, G., 2008. Information Systems Offshoring: Research Status and Issues. *MIS Quarterly*, 32(2), pp.205–225.
- Kotlarsky, J. & Oshri, I., 2005. Social Ties, Knowledge Sharing and Successful Collaboration in Globally Distributed System Development Projects. *European Journal of Information Systems*, 14(1), pp.37–48.
- Krishna, S., Sahay, S. & Walsham, G., 2004. Managing Cross-Cultural Issues in Global Software Outsourcing. *Communications of the ACM*, 47(4), pp.62–66.
- Leinonen, P., Järvelä, S. & Häkkinen, P., 2005. Conceptualizing the Awareness of Collaboration: A Qualitative Study of a Global Virtual Team. *Computer-Supported Cooperative Work*, 14(4), pp.301–322.
- Liang, P., Avgeriou, P. & Clerc, V., 2009. Requirements Reasoning for Distributed Requirements Analysis Using Semantic Wiki. *Proceedings of the 4th IEEE International Conference on Global Software Engineering*, 2009, Ireland, Limerick, pp.388–393.
- Lonsdale, C. & Cox, A., 2000. The Historical Development of Outsourcing: The Latest Fad? *Industrial Management and Data Systems*, 100(9), pp.444–450.

- Luff, P., Hindsmarsh, J. & Heath, C., 2000. *Workplace Studies: Recovering Work Practice and Informing System Design*. United Kingdom, Cambridge University Press.
- Mahadevan, J., 2011. Power/Knowledge in Postcolonial Settings: The Case of IT Bangalore. *Interculture Journal*, 10(13), pp.61–81.
- Majchrzak, A., Rice, R.E., Malhotra, A. King, N. & Ba, S., 2000. Technology Adaption: The Case of a Computer-Supported Inter-Organizational Virtual Team. *MIS Quarterly*, 24(4), pp.569–600.
- Matthiessen, S., Bjørn, P. & Petersen, L.M., 2014. “Figure Out How to Code with the Hands of Others”: Recognizing Cultural Blind Spots in Global Software Development. *Proceedings of the 17th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 2014, Baltimore, Maryland, USA, pp.1109–1119.
- Metiu, A., 2006. Owing the Code: Status Closure in Distributed Groups. *Organization Science*, 17(4), pp.418–435.
- Mockus, A. & Weiss, D.M., 2001. Globalization by Chunking: A Quantitative Approach. *IEEE Software*, 18(2), pp.30–37.
- Montoya-Weiss, M.M., Massey, A.P. & Song, M., 2001. Getting It Together: Temporal Coordination and Conflict Management in Global Virtual Teams. *Academy of Management Journal*, 44(6), pp.1251–1262.
- Nardi, B., 2005. Beyond Bandwidth: Dimensions of Connection in Interpersonal Communication. *Journal of Computer-Supported Cooperative Work*, 14(2), pp.91–130.
- Nardi, B., 2010. *My Life as a Night Elf Priest*, USA, University of Michigan Press.
- Neale, D.C., Carroll, J.M. & Rosson. M.B., 2004. Evaluating Computer-Supported Cooperative Work: Models and Frameworks. *Proceedings of the 2004 ACM Conference on Computer-Supported Cooperative Work*, 2004, Chicago, Illinois, USA, pp.112–121.
- Noll, J., Beecham, S. & Richardson, I., 2010. Global Software Development and Collaboration: Barriers and Solutions. *ACM Inroads*, 1(3), pp.66–78.
- Olson, G.M. & Olson, J.S., 2000. Distance Matters. *Human-Computer Interaction*, 15(2), pp.139–178.
- Olson, J.S. & Olson, G.M., 2013. Working together Apart: Collaboration over the Internet. In Carroll, J.M. eds., *Synthesis Lectures on Human-Centered Informatics*, Morgan & Claypool Publishers.
- Patil, S. Kobsa, A., John, A. & Seligmann, D., 2011. Methodological Reflections on a Field Study of a Globally Distributed Software Project. *Information and Software Technology*, 53(9), pp.969–980.
- Pedersen, M.H. 2012. Ansættelser i udlandet fører til færre, men højere kvalificerede arbejdspladser i Danmark. Dansk Industri: Organisation for Erhvervslivet 2011.
- Pedersen, R. 2011. Prosa: Dette firma er ansvarligt for danske it-fyringer. *Computerworld.dk*
- Pinelle, D. & Gutwin, C. 2003. Designing for Loose Coupling in Mobile Groups. *Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work*, 2003 Sanibel Island, Florida, USA, pp.75–84.
- Prikladnicki, R., Audy, J.L.N. & Everisto, R., 2003. Global Software Development in Practice: Lessons Learned. *Software Process Improvement and Practice*, 8(4), pp.267–281.
- Randall, D., Harper, R. & Rouncefield, M., 2007. *Fieldwork for Design*, United Kingdom, Springer-Verlag London.

- Ravishankar, M.N., Pan, S.L. & Meyers, M.D., 2013. Information Technology Offshoring in India: A Postcolonial Perspective. *European Journal of Information Systems*, 22, pp.387–402.
- Redmiles, D., Van Der Hoek, A. & Al-Ani, B., 2007. Continuous Coordination—A New Paradigm to Support Globally Distributed Software Development Projects. *Wirtschafts Informatik*, 49(1), pp.28–38.
- Sarma, A., & van der Hoek, A., 2002. Palantír: Increasing Awareness in Distributed Software Development. Proceedings of the ICSE 2002 International Workshop on Global Software Development, 2002, Orlando, Florida, USA, pp.28-32.
- Schmidt, K., 2011. *Cooperative Work and Coordinative Practices*, Springer Science & Business Media.
- Schmidt, K., 2002. The Problem with “Awareness”. *Journal of Computer-Supported Cooperative Work*, 11(3), pp.285–298.
- Schmidt, K. & Bannon, L., 1992. Taking CSCW Seriously: Supporting Articulation Work. *Journal of Computer Supported Cooperative Work (CSCW)*, 1(1-2), pp.7–40.
- Schmidt, K. & Simone, C., 1996. Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design. *Computer-Supported Cooperative Work*, 5(2-3), pp.155–200.
- Steinfeld, C., Jang, C.-Y. & Pfaff, B., 1999. Supporting Virtual Team Collaboration: The TeamSCOPE System. *Proceedings of the Conference on Supporting Group Work*, 1999, Phoenix, Arizona, USA, pp. 81–90.
- Steinmacher, I., Chaves, A.P. & Gerosa, M.A., 2013. Awareness Support in Distributed Software Development: A Systemic Review and Mapping of the Literature. *Journal of Computer-Supported Cooperative Work*, 22(1-2), pp.113–158.
- Strauss, A. & Glaser, B.G., 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research*, United States, Rutgers.
- Strauss, A.L. & Corbin, J.M., 1990. *Basics of Qualitative Research*, Sage Publications, Inc.
- Suchman, L., 1995. Making Work Visible. *Communications of the ACM*, 38(9), pp.56–ff.
- Søderberg, A.-M. & Holden, N., 2002. Rethinking Cross Cultural Management in a Globalizing Business World. *International Journal of Cross Cultural Management*, 2(1), pp.103–121.
- Søderberg, A.-M., Krishna, S. & Bjørn, P., 2013. Commitment, Trust and Cultural Sensitivity in Strategic Partnerships. *Journal of International Management*, 19(4), pp.347–361.
- Šmite, D., Wohlin, C., Gorschek, T. & Feldt, R., 2010. Empirical Evidence in Global Software Engineering: A Systematic Review. *Empirical Software Engineering*, 15(1), pp.91–118.
- Treinen, J.J. & Miller-Frost, S.L., 2006. Following the Sun: Case Studies in Global Software Development. *IBM Systems Journal*, 45(4), pp.773–783.
- Tyre, M.J. & Orlikowski, W.J., 1994. Windows of Opportunity: Temporal Patterns of Technological Adaptation in Organizations. *Organization Science*, 5(1), pp.98.
- van Marrewijk, A., 2010. Situational Construction of Dutch-Indian Cultural Differences in Global IT Projects. *Scandinavian Journal of Management*, 26(4), pp.368–380.
- Walsham, G., 1995. Interpretive Case Studies in IS Research: Nature and Method. *European Journal of Information Systems*, 4, pp.74–81.

- Watson-Manheim, M.B., Chudoba, K.M. & Crowston, K., 2002. Discontinuities and Continuities: A New Way to Understand Virtual Work. *Information Technology and People*, 15(3), pp.191–209.
- Whittaker, S. & Schwarz, H., 1999. Meetings of the Board: The Impact of Scheduling Medium on Long Term Group Coordination in Software Development. *Computer-Supported Cooperative Work*, 8(3), pp.175–205.
- Ybema, S. & Byun, H., 2009. Cultivating Cultural Differences in Asymmetric Power Relations. *International Journal of Cross Cultural Management*, 9(3), pp.339–358.

8. Main Contribution

The main part of this dissertation consists of four papers. They will be presented in the following order:

Paper No. 1: Jensen, R. E., & Bjørn, P. (2012). Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds. *Presented at the 10th International Conference on the Design of Cooperative Systems (COOP)*, May 2012, Marseilles, France.

Paper No. 2: Jensen, R. E., & Nardi, B. (2014). The Rhetoric of Culture as an Act of Closure in a Cross-National Software Development Department. *Presented at the European Conference on Information Systems (ECIS)*, June 2014, Tel Aviv, Israel.

Paper No. 3: Jensen, R. E., & Bjørn, P. (2014). Demonstrating Commitment in Practice: Tracing & Gaining Attention. *Submitted to the Journal of New Technology, Work and Employment*.

Paper No. 4: Jensen, R. E. (2014). Why Closely Coupled Work Matters in Global Software Development. *Presented at the International ACM Conference on Supporting Group Work (GROUP)*, November 2014, Sanibel Island, Florida, USA.

Paper Nos. 1, 2 and 4 have already been peer-reviewed and accepted for publication. Paper No. 3 is under review for the Journal of New Technology, Work and Employment.

The next paper to be presented is paper 1

Jensen, R. E., & Bjørn, P. (2012). Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds. *Presented at the 10th International Conference on the Design of Cooperative Systems (COOP)*, May 2012, Marseilles, France.

Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds

Rasmus Eskild Jensen and Pernille Bjørn

IT University of Copenhagen, Technologies in Practice, Rued Langgaards Vej 7, 2300 Copenhagen S, Denmark, email: [raej, pbra]@itu.dk

Abstract This study reports the results of a workplace study of globally distributed software development projects in a global software company. We investigated cultural complexities as social worlds and sought to understand how differences in social worlds between geographically distributed developers become salient in their everyday interactions. By analysing both interviews and observations we identified two types of situations where social worlds become salient in the everyday interactions between developers working at different geographical locations: 1) the divergence of concept and meaning and 2) the convergence of concept but divergence of meaning. We argue that these situations are grounded in social worlds and pose a challenge to work practices in the form of miscommunication and misinterpretation of shared tasks.

1 Introduction

Working in globally distributed teams is increasingly becoming the norm for many large international organizations. Globally distributed work settings are malleable and allow work to be transferred across organizational, national, and cultural boundaries [1], which is attractive for organizations involved with flexible and transferable work like software development. Despite the numerous benefits of globally distributed work settings, there are challenges to coordinating work across sites, including establishing common ground [2], creating suitable work practices [3], and overcoming cultural differences [4]. One key challenge for global software development (GSD) concerns communicating and interpreting implicit knowledge, which is not easily shared out of context [5]. Implicit knowledge is socially embedded within work practices and it is not easily shared

across contexts. Communication requires the development of common ground, and common ground is established through grounded processes [6]. When people collaborate and communicate across social and geographical boundaries, such as languages, organizations, and national borders, the risk of misunderstanding and misinterpretation is high.

Investigating communication complexities in geographically distributed situations, CSCW researchers have examined how people with different national cultures interact with each other and apply different types of media [7-9]. Such studies refer to intercultural communication in terms of national culture, with many referring to Hofstede's different dimensions of understanding the disparities between, for example, Western and Asian countries [10]. However, this perspective on culture as a stable entity has recently come under fire, and it has been suggested that culture should instead be investigated as a social construction between people [11]. We join other CSCW researchers [5] in taking a practice-based approach to examining the cultural complexities in GSD by dealing with the issue of culture as part of practice rather than a stable construct based on nationality. GSD is an interesting area to investigate geographically dispersed collaboration because this type of work comprises closely coupled collaborative tasks and, as such, requires a lot of communication.

While we agree that culture beneficially can be understood as a social construct, we discovered particularly complex and pertinent aspects of communication in our empirical case. These aspects can be linked to the society in which the participants are situated and, as such, are related to culture. This observation made us wonder whether we could think about culture as part of practice without submitting to a national culture framework and yet still take the incidents related to society into account. In this study, we suggest that one way to capture the national perspective on culture without submitting to the categories of Hofstede is to think in terms of social worlds. Investigating the communication within GSD, we therefore asked: How can we identify situations where the differences in social worlds between geographically distributed developers become salient in their everyday interactions?

We report on an ethnographic study of GSD practices within an organization that develops IT systems for Danish customers, with developers dispersed across the Philippines and Denmark. The way people work together in GSD is not always apparent. Too often assumptions are made about the task without examining the underlying implicit knowledge embedded within the task. By making visible situations where differences between developers' social worlds are pertinent to the task at hand, we can better understand how social worlds affect collaboration within GSD.

In the following we start by presenting related work on GSD and communication in virtual teams. Then we present the methodology, our empirical case, and the study results. Finally we discuss our empirical observations and offer conclusions.

2 Theoretical Framework

Aside from cost-savings and despite increased collaborative challenges, the key motivators for developing software across geographical locations are interest in leveraging knowledge diversity, exploiting knowledge capabilities, and scaling activities [3]. Some of the collaborative challenges in GSD identified through ethnographic studies include awareness of distributed collaborators [12], coordination [13], and organizational learning [14]. While these are all important, several researchers have pointed to communicating and interpreting the requirement specifications of the IT system under development as a key challenge [e.g. 15,16].

One of the visions for GSD in any given project is to achieve a shared understanding of the system requirements across the various local sites that are part of the project. System requirements are a key artefact of collaborative practices in all types of software development, and they are used to guide, negotiate, coordinate, and communicate about the tasks shared between developers. However, communication is also considerably more difficult across geographical distances because many details need to be made explicit, yet knowledge is often taken for granted. Empirical workplace studies of GSD have shown that the interpretation of system requirements often causes problems and, in some cases, delays projects or even reduces the quality of the final product [16,17].

One strategy for solving this problem is process standardization, stipulating explicit and detailed requirements. Many large global IT companies (e.g., Infosys and TCS) have chosen this strategy. However, recent studies of GSD question the standardization approach because it restricts a company's flexibility and agility, which may be its core competencies [e.g. 3]. If standardization alone is not the solution, we need to find alternative strategies for dealing with the interpretation of inexplicit system requirements created using taken-for-granted knowledge and background assumptions.

2.1 From Culture to Social Worlds

To investigate how taken-for-granted knowledge and background assumptions affect communication concerning system requirement specifications in GSD, we need a theoretical perspective from which we can examine collaborative practices across cultural boundaries. We define culture as a "reference framework, which stipulates roles and interpretations, and which is dynamically negotiated by the actors in the course of their daily work" [5, p. 20]. This view of culture comprises multi-layered assemblages of intertwined practices, values, beliefs, and attitudes that cannot be isolated or directly examined. It includes lived experiences that guide people's behaviour and attitudes, which consist of unarticulated and taken-for-granted knowledge and beliefs [18, p. 229]. In this way, culture serves as implicit directions shaping the interpretation of events. In a collaborative situation

culture operates as a filter through which collaborators can observe and interpret the actions of others [19, p. 133]. While culture as an isolated factor in collaborative practice, we can study culture by examining the manifestations of culture in practice. This approach to studying culture entails investigating practices, artefacts, and activities as they emerge within GSD with the focus of identifying situations where particular cultural aspects are pertinent for interpreting the situation. However, the question remains how to address cultural aspects when investigating the practical circumstances in GSD.

In collaborative practices, cultural differences are most often invisible until a communication breakdown occurs. Communication breakdowns in geographically distributed teams appear to take place within a shared meaning context comprising three levels: work practice, organization practice, and life world practice [18, p. 231]. Importantly, although a communication breakdown may appear at the work practice level, it might be grounded in either the organizational or the life world level. Thus, we are focused on identifying situations in which communication challenges experienced as part of the work practice level are grounded in the life world level. We speculate that essential miscommunications derived from the life world level are grounded in the social worlds that the participants have grown up and lived their lives in. We propose that the meaning participants assign to particular situations or their understanding of a common task is dependent on the background knowledge they have internalized as part of living in particular social worlds [20]. Social worlds can be described as the institutions and notions about society that is shared among a larger population of people. The social worlds influence our perception and understanding of particular incidents we encounter, including communicating with colleagues who are globally dispersed from us.

By investigating situations where social worlds become salient in the everyday work practice we hope to conceptualize and understand how cultural differences affect communication between geographically dispersed participants involved in GSD.

3 Methods

To answer the research question, we chose workplace studies as our methodological approach [21]. Workplace studies seek to investigate and observe the world as it is and try to understand how people act in the world, making it a suitable approach for investigating communication between practitioners during their everyday interactions. The focus of our case study was GlobalSoft, a GSD company of Danish origin with 1500 employees. Typically clients contact GlobalSoft with particular needs for a new IT system; GlobalSoft also answers public calls for tender specifying the requirements for proposed IT systems. Regardless of how the connection between GlobalSoft and the client is created, all projects begin with key negotiations about the scope of the project. In most cases GlobalSoft negotiates directly with the client, with little or no involvement from offshore partners. Once

the project scope is defined, it is divided into tasks, some or all of which are sent to the Philippines, depending on the project. The Filipino department's only input on the project prior to this is when they are asked to do a task proposal estimating the number of hours required for at given task. One of the key documents in this process is the requirement specification, which is meant to define the scope of the entire software project.

As part of a large research project on GSD, we initiated a workplace study with GlobalSoft in November 2010 (study ongoing). In total, three researchers conducted 22 audio-recorded interviews (14 in Denmark, 8 in Philippines) lasting 30–60 minutes (average 50 minutes). Practices were observed in Manila, Philippines, for approximately 180 hours over three periods (December 2010, July 2011, November 2011). In Copenhagen, Denmark, observations of a particular project were conducted for approximately 80 hours. Four workshops (2 in Denmark, 2 in Philippines) were conducted, and various documents and presentations were collected. Employees at many different organizational levels were interviewed, allowing us to compare perceptions of the corporate vice president of GSD with those of the developers.

4 Results

We documented several incidents where the challenges developers experienced could not be explained by normal communication issues like trust [2] or social context [22]. Instead, these incidents were related to the local social worlds of the different participants. Here we present four examples of situations where the differences in social worlds between geographically distributed developers become salient in their everyday interactions: Prescriptions and pharmacy, CVR and CPR, food and health inspections, and retirement plans. We also present two examples of social worlds at work: Inventory facility management system and children with special needs.

4.1 Diversities in Social Worlds

4.1.1 Prescription and Pharmacy

Our data suggest that the Danish employees often experienced their Filipino colleagues misinterpreting or misunderstanding the intended meaning of the requirement specification. The requirement specification contains overall descriptions of all the tasks for a given project. But the descriptions of the tasks are often part of a

predefined context, which is the result of assumptions that are embedded in the requirement specification and which can cause misinterpretations and misunderstandings. A Danish manager from GlobalSoft explained the situation:

[The Danes] should also understand that they [the Filipinos] may not recognize everything. That they spend time talking about what a prescription is. And what a pharmacist is. (Manager)

The manager quoted above spoke about how different understandings of a concept can influence the project. In this example he mentioned prescriptions and pharmacists as concepts that were perceived differently by the Filipinos. In Denmark, all pharmacists have undergone 5 years of university training and are strictly governed by official authorities. All prescriptions are sorted in IT systems and are efficiently monitored by the authorities. Doctors authorize prescriptions after patient consultations and submit them to a general database that all pharmacists can access. Patients can then go to the nearest pharmacy and collect the medicine. The Danish manager recognized that the Filipino employees might not fully comprehend the complexities of how prescriptions and pharmacies are integrated and administrated by the officials in the Danish system. It therefore became essential to talk about what a prescription or a pharmacist was to develop a common understanding of these concepts. The manager was describing the challenge of understanding the local context of Danish pharmacies and prescriptions - a challenge grounded in the social worlds between the Danish and Filipino employees.

4.1.2 CVR and CPR

A Danish IT architect described the challenge of communicating possible differences in social worlds:

But there are also some things we take for granted. I do not need to tell a Danish programmer what a CPR number is, or a CVR number, or many other things, because we take it for granted.... But when we are speaking with the Filipinos... then it is not certain that they have the same knowledge. Such cultural issues, which are something we [the Danes] all know about, are not known outside the borders of the country. And that can easily cause misunderstandings. (IT-architect)

The IT architect described the Danish Central Personal Registry number (CPR), which is unique to Denmark. Every Danish citizen is given a CPR number at birth used as identification for every Danish citizen. In some ways it is equivalent to a social security number, except that it is not optional; everyone must have one. All interactions between the public and the municipalities or the government such as healthcare, taxes, day care, and education, are managed through a CPR number. Thus, the CPR number is an integral part of the social world in Denmark. However, the extensive use of the CPR registration is viewed as controversial in other countries, such as United States, where it is often perceived as unnecessary governmental control of citizens in a democratic country. This example illustrates that outside of Denmark, the concept of the CPR is not fully comprehended, yet the

development of IT systems for Danish institutions will require a comprehensive understanding of the concept and the criteria surrounding it. The Central Company Registry number (CVR) is used for registered companies of a certain size. Companies with revenue of 50,000 Danish kroner or more have to register for a CVR number in a centralized database called Virk.dk and cannot function legally without one. According to the IT architect, there is considerable taken-for-granted knowledge about Danish society that is not easily communicated. This became evident when the Danes were trying to communicate the meaning of the CPR number to their Filipino colleagues, because they lack a shared context.

4.1.3 Food and Health Inspections

Another challenge we documented involved communicating the use context for a particular IT system. In 2001, the Danish government initiated a public food inspection program in an attempt to secure the health of Danish citizens. Inspectors travel the country regularly visiting restaurants and giving them a general hygiene rating. These ratings correspond to various smiley faces, where the cleanest restaurants get a full smiley face and the less clean get a sad face. A Danish project manager had the following experience when working with his Filipino colleagues on a scheduling system for governmental food and health inspectors:

Yes it [the project] is about route schedules. In this case, an inspector who should visit two restaurants in the course of one day, and each visit should take approximately one hour. To a Dane, this is clearly a mistake, because what was the inspector going to do for the rest of the day? But for the Filipinos, (...) well, they did not relate it [the product] to the application. (Project manager)

The project manager quoted above presented an example of how different perceptions of a concept caused a misunderstanding. The developers in the Philippines were coding a scheduling system where data (i.e., number of inspectors and number of restaurants to be checked) were entered and the system generated route schedules. However, at some point an error occurred in the system, resulting in inspectors being assigned to only two visits a day. The Filipino developers tested the IT system using various use-case scenarios and did not find this error. They did not realize that two inspections a day translates into a two-hour workday, which, by Danish standards, is not a sufficient use of resources. To the Danes, this was clearly a mistake. But, according to the Danish project manager, the mistake was grounded not in the Filipino developers' lack of ability but rather in a fundamental lack of understanding of the IT system and how it would be used in Danish society. We suggest that the Danish developers identified the error not based on a superior understanding of the requirement specification but due to a fundamental understanding of the Danish society. As the manager stated, the Filipinos did not understand the use context and therefore could not relate the IT system to the use situation. Not being able to relate the task to a given context is a challenge that both the Filipinos and the Danes are aware of. Yet it still remains a source of miscommunication.

4.1.4 Retirement Plans

The following example illustrates the challenge of interpreting system requirements with unknown terminology, in this case, while developing an IT system for retirement plans. A Danish manager explained:

We had a couple of discussions regarding the retirement concept (efterløn) and your public pension age (folkepensionsalder), which was misunderstood. They [the Filipinos] had understood it in one way and we had another idea of the concept. And this meant that our testing did not match, and at some point the client got involved, because the correction of one error resulted in new errors, basically because they had corrected more than they were supposed to. And it was all caused by this confusion of concepts.
(Advanced Project Manager)

In Denmark, everyone gets a public pension at the age of 65, but many also have a privately funded pension. On top of that, many Danes are part of a public retirement fund called “efterlønsordningen,” which is a supplement to the national pension plan aimed specifically at blue collar workers. This voluntary plan is intended to retire the older generation and create demand for younger workers by allowing workers with physically demanding jobs to retire at 60 instead of 65. In the Philippines they have a social security system where the employer and employee each make monthly contributions based on the employee’s monthly wage. The contributions depend on the salary, and the employer matches the amount contributed by the employee. This is a mandatory minimum that legally obliges Filipino workers and companies to create retirement funds. Workers become eligible for their pension around the age of 55–60, and there is no public pension plan. Furthermore, Filipino law requires companies to pay one month of salary for every year of service to employees who have been with the company for at least 10 years when they turn 55.

These very different retirement plans are built on the particular social worlds of their respective countries. Thus, constructing an IT system to manage retirement plans requires significant knowledge of the social systems in the given country. This knowledge is not easily transferred between developers; it requires considerable communication not only about the system requirements but also about the society in which the IT system will be implemented.

According to the Danish manager, the differing pension systems led to different interpretations of the concept, which led to errors in the IT system that were eventually detected by the Danish testers. This example illustrates the invisibility of different background assumptions and taken-for-granted knowledge. The manager saw a strong relationship between the problems they experienced in the project and the lack of a common vocabulary for the project because the developers shared the concept, but not the meaning behind the concept.

4.2 Social Worlds at Work

4.2.1 Inventory Facility Management System

When the Filipino employees are given a task, it is often in the form of a paper-based requirement specification. We observed two Filipino project managers discussing a task proposal sent from Denmark. They discussed how to determine the scope of the task and how to estimate the number of hours required to complete the task. During the discussion, the developers turned to us and reflected on how difficult and prone to misinterpretation this activity is:

Project Manager: Then there are these requirements like: There should be a...

Interviewer: Bruttoliste?

Project Manager: Yeah, see, we can't understand that. What is that? So what kind of list is that? My assumption is, like, gross list. I am not sure, but based on our assumptions, bruttoliste is like a contract list. So, I am not sure how we can use that term.

The Filipino project manager did not understand the term “bruttoliste”, a Danish word related to calculating inventory. If the word is directly translated, it means a gross list, but, according to the Filipino manager, it could be interpreted as a contract list. The project manager tried to relate the concept to its meaning, which is key to understanding the IT system they are going to create. The challenge was further complicated in this case because the Danish developers did not translate the word bruttoliste. Forgetting to translate is a common mistake in GlobalSoft, despite assumptions about the concept being used to create a sense of meaning for the proposed IT system. To overcome this challenge, the Filipino project managers created a list of assumptions showing how they have interpreted the task description.

There are a lot of assumptions in this project. There are, like, 16 assumptions. Yes, it is quite a lot. Because of the requirements. Did you see the requirements? How would we work on that? (Project manager)

The Filipino project manager was clearly frustrated with the difficulty making sense of the requirements. Creating assumptions was their way of trying to overcome the uncertainties in the task proposal. In this case, they had to create a list of 16 assumptions to estimate the number of work hours for the project. If any of these assumptions are wrong, then a new estimation of time and resources is required. A Filipino project manager continued discussing the gross list IT system:

(...) this facility asset thing...we are not used to those kinds of systems. So, as mentioned before, it basically comes down to the domain knowledge frustration. So, we don't have any domain knowledge about this system. (Project manager)

The manager explained they do not always fully understand the requirement specification. She referred to this problem as “domain knowledge frustration.”

4.2.2 Children with Special Needs

It is not only the lack of a common vocabulary that creates challenges for communication. In GlobalSoft, we also saw examples where differences in social worlds became pertinent for the collaboration. While working to create an IT system for children and youth with special needs, a Danish team leader experienced both a lack of understanding and scepticism as a result of different social worlds:

We had a project about a system for handling children and youth with special needs. And we had negotiated the scope, but they [the Filipinos] never realized that this was a big project of great importance because, in their eyes, they believed that, frankly, you cannot allocate that much money for these activities. They [the children] should be able to look after themselves. Because this is how they do it in their [Filipino] society. (Team Leader)

In Denmark, the social welfare system is investing heavily in children and youth with special needs ranging from learning difficulties to severe physical disabilities. It is a high-priority issue that has general support from most political parties. However, the Danish team leader indicated that the Filipino developers could not grasp the importance of the project. In this example, social worlds became salient. Even after the scope of the project was negotiated, the project leader found it difficult to convince the Filipino developers why the project was important. They showed disbelief that so much money could be allocated to children with special needs. In Filipino society, such children have to look after themselves and would not be supported by the government. The team leader described further:

So it was linked as a central solution to an important project and we had money prioritized for these things, right? But they [the Filipinos] never really took it seriously. Because, in their context, this seems like a completely ridiculous way to spend money. (Team Leader)

This quote illustrates how social worlds become salient in global work. The Danes felt they had a solid project with a straightforward solution and that overall the project was important and highly prioritized by the client—the Danish government. But the Filipino developers were sceptical about the project. They struggled to understand the willingness to spend so much money on children with special needs. Because the Filipino employees had trouble relating this project to a meaningful situation, they were, according to the manager, not able to collaborate in a serious manner.

No, this can be a real challenge because sometimes they [the Filipinos] find it very hard to understand...to understand what really concerns people in Danish society, and why many things can be important in Denmark when they do not understand them at all. (Team Leader)

According to the team leader, the real challenge is a basic lack of insight into the different social worlds embedded in the geographically distributed teams. The Filipino developers have trouble understanding the social contexts in which the IT systems will be applied. This lack of contextual knowledge is an obstacle for the collaboration and may increase the risk of communication breakdowns.

5 Discussion

We have been investigating situations in practice where differences in social worlds between Danish and Filipino developers became salient. Our empirical observations demonstrated how social worlds became a pertinent part of collaborative work in relation to concept and meaning. We set out to identify the types of situations in which the differences between the developers' social worlds were visible and affecting the collaboration. We propose two general situations where social worlds become salient: 1) divergence in concept and meaning and 2) convergence in concept but divergence in meaning. We do acknowledge that the categories "meaning" and "concept" might be a simplistic model for illustrating complex situations of matches and mismatches, but we propose it as a way to unpack the concept of culture.

5.1 Divergence in Concept and Meaning

We saw several examples where concepts essential for the design of the IT system were not part of the social world of the remote site, including the CPR number, a key part of the structure of Danish society that is thus crucial for IT systems designed for the Danish government. Social security numbers are used in the Philippines, but there are clearly differences between Danish CPR numbers and Filipino social security numbers. The divergence between social worlds in Denmark and the Philippines is quite evident, so the parties are aware of the differences. They know they do not share knowledge and must therefore make extra efforts to explain the concept as well as the meaning behind the concept to the remote site. In situations where concepts relevant for the interpretation of the system requirements are localized in the social world of one location, and where the concept is not part of the shared meaning context of the other location, identifying the issue and resolving communication breakdowns is a process of explaining, negotiating, and creating a shared meaning context [18].

In the food and health inspector example, we observed the Filipino developers' failure to relate the IT system to the use context, leading to errors that were easily identified by the Danish testers. In such a situation it is crucial that the remote party identify and question the assumptions embedded within the system requirements, as others have argued [17,16]. We suggest that situations of divergence in concept make it easier to identify the divergence in meaning of the concept because in the remote location, the concept does not exist. In these situations, reinterpretation of the meaning behind the concept is not needed.

5.2 Convergence in Concept but Divergence in Meaning

As we have shown, high diversity in domain vocabulary across sites creates extra work that is critical to creating a shared meaning context for the project. In high diversity situations, developers working across sites might be aware of the risk of misunderstandings, and thus will use considerable resources translating domain-specific knowledge in documents like the requirement specification and the product description. It is relatively easy to identify instances of divergence in concept because the lack of a shared vocabulary is obvious. However, our analysis revealed a different type of situation where the local social world was evident—situations where the concept is shared across locations but has different meanings for the different social worlds, such as the case of the retirement system. The retirement systems in Denmark and the Philippines appeared quite different and were dependent on the social world of each country. We label this a situation where there is convergence in concept but divergence in meaning across different social worlds. Differences in social worlds prove challenging for the design and development of IT systems in cases of a divergence in meaning of key concepts used in both local contexts. Both Denmark and the Philippines have an interpretation of the concept of a pension plan, and the concepts have similarities across locations, such as similar retirement age. There are, however, distinct differences in how the two societies have structured their pension systems, but these differences may not be immediately apparent to the development teams.

The lack of visibility of different parties' interpretations of a common concept makes it difficult to detect this source of misunderstanding. We saw this in the retirement system example, where crucial errors were not detected early by the Filipino testers and were only later discovered by the Danish testers. This suggests that communication breakdowns caused by diversity in meaning of shared concepts are more likely to happen at later stages because participants may perceive a "false" sense of common ground, making the lack of shared understanding harder to identify. Common ground occurs when parties share knowledge and know they share it [2]; however, developing common ground requires a grounding process where the parties rearrange their knowledge according to each other's utterances [6]. In cases of convergence in concept but divergence in meaning, detecting a lack of common ground is difficult and, in some situations, even impossible. In many cases, the lack of common ground will not appear until technical decisions based on false assumptions become manifested in the IT system. This kind of miscommunication is more costly for the company because the cost of fixing errors rises exponentially as the product reaches the delivery deadline. It is essential that the remote party not only identify and question the assumptions embedded within the system requirements, but also that they are aware of the divergence in meaning of what might seem to be a shared understanding of key concepts. We believe identifying situations of convergence in concept but divergence in meaning is difficult but critical to reducing the risk of miscommunication throughout the development process. We argue that re-interpreting the meaning behind the concept is required for developers to establish a shared understanding of the development of

IT systems. Yet, because of the initial shared understanding of the concept, development teams tend not to allocate resources to communicating its meaning and thus fail to address the challenge of divergence in meaning.

You may also wonder about situations of convergence in concept and meaning. We detected no such cases when examining our data for situations where differences in social worlds between sites became salient for communication. We might assume that in situations of convergence in both concept and meaning, no differences in social worlds exist, and thus they are not problematic. Perhaps our empirical material revealed no such situations because they do not result in communication breakdowns, and thus our respondents did not identify these instances as challenges for collaboration in GSD.

We have argued that social worlds can become salient as either divergent in concept and meaning or convergent in concept but divergent in meaning. We propose that these social worlds pose a risk for miscommunication between developers, and we have observed examples of these miscommunications in practice. In the inventory facility management example, where the Filipino project leaders found it challenging to comprehend the IT system's use context, these challenges lead to frustration because a list of assumptions had to be created as a way to address these challenges. Workaround creation in GSD processes has also been noted by other researchers [3]. In such situations, social worlds become salient and project leaders create ways to work around them.

The development of an IT system for children with special needs was another example of social worlds at work. In this case, differences in social worlds led to scepticism among the developers across locations and created a challenge for the collaboration. Thus, the lack of implicit directions for shaping the interpretation of events hindered the developers' ability to enact shared meaning [18].

Based on our observations of social worlds and their impact on collaboration, we suggest the need for further research into understanding how divergence and convergence in concept and meaning affect collaborative work in GSD. We propose that divergence in concept and meaning is easier for participants to identify and comprehend and will therefore often be present early in the development process. In contrast, convergence in concept but divergence in meaning is harder to identify and more complicated to comprehend because it requires reinterpretation of familiar concepts and we suggest that these situations will occur later the development process.

6 Conclusion

In this study, we present an analysis of a work place study in a global software development company. We set out to investigate how to identify situations where the differences in social worlds between geographically distributed developers become salient in their everyday interactions. In our analysis, we identified two

types of situations where social worlds became pertinent: 1) situations of divergence in concept and meaning and 2) situations of convergence in concept but divergence in meaning. While we acknowledge that the conceptualization of concept and meaning might be somewhat simplistic, we believe this to be a first step into unpacking culture as part of collaboration. Based on our empirical findings we have argued that by identifying and describing these situations, we can better understand how and why communication breakdowns occur in intercultural collaborative work practices.

Acknowledgements

This research has been funded by the Danish Agency for Science, Technology and Innovation under the project "Next Generation Technology for Global Software Development", #10-092313.

References

1. Majchrzak A, Rice RE, Malhotra A, King N, Ba S (2000) Technology Adaptation: The Case of a Computer-Supported Inter-Organizational Virtual Team. *MISQ* 24 (4):569-600
2. Olson GM, Olson JS (2000) Distance Matters. *Human-Computer Interaction* 15:139-178
3. Avram G, Bannon L, Bowers J, Sheehan A (2009) Bridging, patching and keeping the work flowing: Defect resolution in distributed software development. *Comput Support Coop Work* 18:477-507
4. Krishna S, Sahay S, Walsham G (2004) Managing cross-cultural issues in global software outsourcing. *CACM* 47 (4):62-66
5. Boden A, Avram G, Bannon L, Wulf V (2009) Knowledge management in distributed software development teams: Does culture matter? Paper presented at the International conference on Global Software Engineering (ICGSE), Limerick, Ireland, 13-16 July
6. Clark H, Brennan S (1991) Grounding in Communication. In: Resnick L, Levine J, Teasley S (eds) *Perspectives on Social Shared Cognition*. American Psychological Association, Washington DC, pp 127-149
7. Massey AP, Hung Y-TC, Montoya-Weiss M, Ramesh V (21) When culture and style aren't about clothes: Perceptions of task-technology "fit" in global virtual teams. In: *GROUP*, Boulder, Colorado USA, 2001. ACM, pp 207-213
8. Nardi BA, Whittaker S, Bradner E (2000) Interaction and outercation: instant messaging in action. Paper presented at the Proceedings of the 2000 ACM conference on Computer supported cooperative work, Philadelphia, Pennsylvania, United States
9. Setlock L, Fussell S (2010) What's it worth to you? The costs and affordances of CMC Tools to Asian and American Users. Paper presented at the Computer Supported Cooperative Work (CSCW), Savannah, Georgia, US
10. Duarte D, Snyder N (2006) Mastering virtual teams: Strategies, tools, and technoques that succeed. John Wiles & Sons Inc, San Francisco, California
11. Söderberg A-M, Holden N (2002) Rethinking cross cultural management in a globalized business world. *International Journal of Cross Cultural Management (CCM)* 2 (1):103-121

12. Souza Cd, Redmiles D (2007) The awareness network: To whom should I display my action? And, whose actions should I monitor. Paper presented at the European Conference on Computer Supported Cooperative Work (ECSCW), Limerick, Ireland
13. Espinosa A, Slaughter S, Kraut R, Herbsleb J (2007) Team knowlegde and coordination in geographically distributed software development. *Journal of Management Information Systems* 24 (1):135-169
14. Boden A, Nett B, Wulf V (2010) Operational and strategic learning in global software development: Implications from two offshoring case studies in small enterprises. *IEEE Software* 27 (6):58-65
15. Herbsleb J (2007) Global software engineering: The future of socio-technical coordination. Paper presented at the Future of Software Engineering (FOSE), Washington, DC, USA,
16. Boden A, Nett B, Wulf V (2009) Trust and social capital: Revisiting an offshoring failure story of a small German software company. Paper presented at the European Conference Computer Supported Cooperative Work (ECSCW'09), Vienna, Austria, September 7-11
17. Herbsleb J, Paulish D, Bass M Global software development at Siemens: Experience from nine projects. In: International conference on software development (ICSE), St. Louis, Missouri, USA, 2005. ACM, IEEE Xplore, pp 524-533
18. Bjørn P, Ngwenyama O (2009) Virtual Team Collaboration: Building Shared Meaning, Resolving Breakdowns and Creating Translucence. *Information Systems Journal* 19 (3):227-253
19. Ngwenyama O, Klein H (1994) An Exploration of the Expertise of Knowledge Workers: Towards a Definition of the Universe of Discourse for Knowledge Acquisition. *Information Systems Journal* 4:129-140
20. Star SL, Griesemer J (1989) Institutional ecology, translations and boundary objects: Amateurs and professionals in Berkeleys museum of Vertebrate Zoology, 1907-39. *Social Studies of Science* 19:387-420
21. Randall D, Harper R, Rouncefield M (2007) *Fieldwork for design: Theory and practice*. Springer, London
22. Jarvenpaa SL, Leidner DE (1999) Communication and Trust in Global Virtual Teams. *Organ Scien* 10 (6):791-815

The next paper to be presented is paper 2

Jensen, R. E., & Nardi, B. (2014). The Rhetoric of Culture as an Act of Closure in a Cross-National Software Development Department. *Presented at the European Conference on Information Systems (ECIS)*, June 2014, Tel Aviv. Israel.

THE RHETORIC OF CULTURE AS AN ACT OF CLOSURE IN A CROSS-NATIONAL SOFTWARE DEVELOPMENT DEPARTMENT

Complete Research

Jensen, Rasmus Eskild, IT University of Copenhagen, Denmark, raej@itu.dk

Nardi, Bonnie, University of California, Irvine, US, nardi@ics.uci.edu

Abstract

Global software teams work on interdependent tasks across geographies, time zones, and cultures. Studies of cross-national software teams report that the main challenges are sharing knowledge, creating trust, and establishing common ground. In this study we examine another challenge—the use of the word “culture” as an exclusionary act of closure. As theorized by Metiu, who builds on Weber, closure occurs when one group seeks to preserve the advantages of their situation relative to another group. We conducted an ethnographic study of a cross-national software department with members in Denmark and the Philippines. We found that “culture” was invoked by employees in the Danish office to explain failed collaborations, but not by employees in the Philippines. Filipino employees formulated other explanations for problems, and actively resisted cultural explanations. We argue that employees in the Danish office used a “rhetoric of culture” as an act of closure, and we focus on technology choices and behavior as they were impacted by closure.

Keywords: Closure, Exclusion, Cross-national software development teams, Rhetoric of culture

1 Introduction

Global software groups working in distributed settings are increasingly common (Malone 2004, Carmel and Tjia 2005, Conchúir et al. 2009, Noll et al. 2010, Ehrlich and Cataldo 2012, Tang et al. 2012). To examine problems of cross-national collaboration, we conducted an ethnographic study of “GlobalSoft” (a pseudonym), a software company based in Denmark that has expanded operations by offshoring to several locations. The findings are part of a larger longitudinal study of cross-national collaboration within a GlobalSoft department with employees in Denmark and the Philippines. The software department we studied had already experienced some of the typical challenges of cross-national work reported in the literature such as sharing knowledge (Cramton 2001), creating trust and social relations (Jarvenpaa and Leidner 1999), and establishing common ground (Olson and Olson 2000). This paper focuses on acts of closure and how they affected work relations and technology use.

Many studies of cross-national groups point to cultural differences as a key challenge in managing projects (see King and Torkzadeh 2008, Kumar and Jha 2010). Asymmetric relationships and subgroups often emerge in such work units (Chand et al. 2009). Group members construct categories to differentiate themselves from one another (Cramton and Hinds 2005, Metiu 2006, Marrewijk 2010). Categorization may be based on “any convenient characteristic...including race, social background, language, religion, and gender” (Metiu 2006). Asymmetries and subgrouping may lead to closure, which was theorized by sociologist Max Weber as competitive behavior that occurs when “one group of competitors takes some externally identifiable characteristic of another group of [actual or potential] competitors...as a pretext for attempting their exclusion” (Weber 1978). Weber noted that acts of closure potentially exclude others from social and economic resources (see also Bottero 2005).

For example, Metiu studied how a group of US developers in a cross-national team secured desirable tasks for themselves by interacting infrequently with their Indian counterparts, exhibiting a generally critical attitude toward them, and often ignoring them. These actions constituted a “strategy of closure” to maintain resources for the US developers (Metiu 2006).

In the cross-national department we studied, the higher status workers were Danish, and lower status workers Filipino. By “high” and “low” status we refer to differences in decision-making power and influence on tasks and work processes. Danish workers were responsible for sales, client relations, developing software architectures, programming, and assigning and scoping work tasks. Filipino workers were responsible for programming modules of the software products Danish managers deemed the least complicated, and for testing. Danish employees evaluated the quality of the work done by Filipino workers, but not vice-versa. In this setting, the “identifiable characteristic” of closure, from the point of view of Danish workers, was *culture*. Earlier studies of cross-national work groups focused primarily on the concept of culture and how to mitigate the problems of cultural differences (Søderberg and Holden 2002, Krishna et al. 2004, Kayan et al. 2006, Diamant et al. 2008, Deshpande et al. 2010). We found that questions of culture revolved not around cultural misunderstandings but issues of power and influence. We contribute to a growing body of literature (Mahadevan 2011, Ravishankar et al. 2013) investigating the role of culture in terms of power relations in cross-national teams. We show how higher status employees sought to protect their tasks and roles by invoking “culture” as the root of collaboration problems. The vague formulation of “culture” was more subtle and ambiguous than, for example, deliberately ignoring someone or being highly critical, as observed in other research (Jarvenpaa et al. 1998, Metiu 2006). We show that the rhetorical use of culture was a clear pattern of behavior, and was used only by employees in the Danish office. As we will discuss, it appears that collaboration with Filipino employees represented a threat to the existing order of work for the Danish employees, and even possible future employment.

Our research concerns *informal* acts of closure. Formal means of closure occur at the institutional level (Weber 1978, Weeden 2002, Bidwell 2013) - for example, policies privileging one group over another. GlobalSoft was organizationally committed to equitable collaboration between national groups. But the imposition of offshoring destabilized work practices for the Danish workers, leading to informal closure. Top management offered the rationale that offshoring would allow the company to remain competitive:

It [offshoring] is driven by the market. We do not—this is very important—we do not relocate jobs to the Philippines or China. We are moving tasks, so in a growing market we will be able to sustain the work force in Denmark, assuming that the people we have are willing and capable of changing their roles and professional profiles. But as I have...said...there is no job guarantee. (Vice president, Denmark)

But many employees in Denmark questioned the rationale behind offshoring. A competitive edge derived from employing Filipino workers is a claim that remains unproven as the company has not conducted a comparative study of the cost difference between work done at the primary location and work done offshore. A recent study found that negative attitudes are likely to emerge when employees see little benefit from offshoring (Zimmermann 2012). Danish employees remained skeptical towards offshoring, and tried to maintain tasks and decision-making power locally through informal acts of closure. Metiu argues that more research is needed regarding “how the actions of group members exclude others [even] when their official organization is committed to the cooperation” (Metiu 2006). We therefore ask: *How are informal patterns of closure enacted in cross-national collaboration?* A contribution of our work is to point to the rhetorical use of the word *culture* as an act of closure in cross-national distributed work, something not yet reported in the literature as far as we know. By applying a sociological perspective to IS offshore studies we also contribute with much needed insights to the cultural and informal management mechanisms (Gregory 2010).

2 Methods

GlobalSoft incorporated in Denmark in 1994. It employs about 1700 people in Denmark, China, Switzerland, the Czech Republic, and the Philippines. Filipino employees originally worked for an independent supplier of programming resources hired by GlobalSoft. In 2009, GlobalSoft bought the supplier, and hired its 85 employees. Employees merged with an existing department in Denmark. Danish employees were generally older and more experienced than Filipino employees who were often hired straight from the universities, and were younger and paid less than their Danish counterparts. Employees collaborated through several forms of mediated communication including email, instant messaging (IM), video meetings, and shared documents.

Data collection took place from December 2010 to October 2013. We studied a single department in a single company, so there were no differences in corporate policy across the Danish and Filipino workers. Study participants were highly professional and worked with expertise within their fields. We conducted on-site observations in Denmark and the Philippines, shadowing employees, participating in video meetings, and observing everyday practices. The data include 28 audio-recorded and transcribed interviews (19 in Denmark, 9 in the Philippines), each about an hour long. We interviewed, observed, and interacted with developers, testers, IT-architects, project leaders, and managers. In this paper we refer to everybody as employees or workers, but we occasionally call out the managerial role. We spent a total of almost four months in the Philippines (December 2010, July 2011, November 2011, and January 2012), and 12 months in Denmark. During data collection, we discussed initial findings with the workers in casual conversation and in official presentations and workshops. Two video-recorded workshops were held in the Philippines. One took place with a group of managers and another with a group of developers. Only employees from the Philippines participated in the workshops. Researchers asked participants to brainstorm about the main challenges in the collaboration and discuss possible solutions to these challenges. We used the presentations and workshops to interrogate and validate our findings. Finally, we examined video recordings of work practice, as well as internal company documents, requirements specifications, and official materials.

We applied an open-ended study design inspired by grounded theory (Strauss and Glaser 1967). Interview transcripts and field notes were systematically read for broad themes and then coded according to the finer-grained themes discussed in the findings. The themes were categorized using a text-analysis tool called TAMS Analyzer where the field notes and interviews were coded creating categories such as “communication_patterns”, “relation_work” or “instant_messaging”. The variety of data collection methods (interviewing, participant observation, workshops, screen recordings) helped establish a grounded understanding of the complexities of cross-national development practices. Analyzing the qualitative data required several iterations of reading, coding, and creating write-ups to connect the themes that emerged (Eisenhardt 1989). We engaged in formal discussions about the themes after reading the data and collectively converged on interpretations, making sure each point was well supported in the data in multiple places to support the reliability and validity of the findings. We presented our work for a group of researchers at an internal research seminar to gather feedback on the initial findings. Many of the findings are based on employee reflections of the collaboration practices, supported by our observations. The rhetorical use of culture became evident in interviews, workshops, and presentations when employees reflected on the collaboration. These reflections took place in the presence of the interviewers or members of their own national group. Culture was rarely mentioned in cross-national conversations.

3 Findings

In the following section we discuss how Danish workers' assertion of cultural difference affected the collaboration. We describe the enactment of closure and the consequences for the work and for the use of collaborative tools. Finally, we analyze the underlying motives for closure in the collaboration.

3.1 Assertion of and resistance to notions of cultural difference

Danish employees often asserted that culture was the underlying reason for failures in the team's collaborative work. For example, a Danish manager said in an interview: *"The largest barriers are often determined by culture."* Another manager told us: *"They [Filipino workers] live in an entirely different world. What can we possibly talk about?"* In another interview, a third Danish manager said: *"These cultural things which we [Danish workers] all know about, well, people from outside our borders do not know these things, and that can easily cause misunderstandings."* In interviews, we asked the Danish managers to elaborate on the meaning of these cultural "things." They would often point to language. For instance, one manager said that culture is a *"a language issue."* Another said:

Ok, maybe I put too much into the word culture. But it is exactly things like language. Is it culture? I do not know. But the fact that you have to speak a second language is definitely a barrier for picking up the phone in the morning.

Here a manager trades on the ambiguity of the word culture. At first it appears that he might mean something very broad, very pervasive. But when asked to explain, he narrows "culture" to language, and remarks how it makes calling workers in the Philippines—to whom he will have to speak in English—difficult. In another interview, a manager said: *"[M]isunderstandings can emerge, and it is not a question of apportioning blame. Instead we have to remember that we do not share the same logical background in both culture and language, right?"* This manager indicated that language could cause misunderstandings, but he also expressed a strong statement of difference arising from cultural logics. In this view, Danish and Filipino workers lacked a "shared logical background" and misunderstandings were thus not surprising.

Another Danish manager said: *"We do not think that all of them [Filipino workers] reveal themselves as they maybe would have done if they were in Denmark with a Scandinavian origin and were not culturally different."* Danish managers saw themselves as members of an outspoken culture typical of "Scandinavia", as opposed to quieter Filipino workers. We noticed that in meetings it was often Filipino managers who asked and fielded questions, while other Filipino workers remained quiet. The Danish manager's statement draws sharp lines between those of Danish/Scandinavian origin and those who are "culturally different." Lost in this characterization is the open discourse of Filipino managers who spoke freely at meetings. Culture was invoked not only as belonging to "Denmark," but also as essentially "Scandinavian," constructing Scandinavia as a locale with specific cultural qualities (such as outspokenness) the Danish workers valued. The concept of culture was used rhetorically to assert a stable state of difference. Such statements expressed sentiments shared by many employees in the Danish office.

Filipino employees, by contrast, resisted cultural interpretations as a reason for miscommunication. When asked about such differences in an interview, one Filipino manager said, *"[Culture] is not as big as I think people say it is. Because a lot of it isn't culture specific."* Instead, the manager enumerated other factors important to collaboration such as geographical distance, time difference, and language. She mentioned language as a problem but did not subsume it under culture as some Danish managers did. In an interview, a Filipino worker pointed to the time difference, saying, *"It's really problems with the time zone."* This worker spoke in specific, non-cultural terms, highlighting a specific condition affecting collaboration.

Filipino workers argued that Danish workers' poor communication practices led to problems. Even a Danish manager who had spent time working in the Philippine office said during a workshop: *"People confuse poor communication with cultural differences. It is very popular to say that it is a cultural issue. I actually do not think there are that many cultural issues."* This manager resisted the "popular" cultural interpretation of collaboration problems. He offered a narrowly scoped claim about communication practices bearing on the problem.

3.2 Attributing cultural differences to explain the challenges of work tasks, everyday practices, and the use of collaborative tools

In this section we discuss specific problematic work tasks managers in Denmark often attributed to cultural differences.

3.2.1 Coordination of time difference

The time difference of seven hours would typically allow an overlap of only about two hours in which both Danish and Filipino employees were at work. The Filipino workers noted that the Danish employees did not always acknowledge the time difference: *“I am actually surprised how many people ask me what time it is here,”* one manager in the Philippine office commented at a workshop. At the workshop another Filipino manager said:

They [Danish employees] sometimes forget that we have a time difference, so it is basically that they email us at 8.00 a.m. Danish time and they want the task done on that same day and it is kind in the afternoon for us, so...

Filipino workers, on the other hand, were well aware of the time in Denmark. They had clocks on their computers as well as clocks on the wall showing the time in Denmark. Danish employees had no indicators of Philippine time in their work environments. A manager in the Philippine office found the “forgetting” of the time difference somewhat frustrating: *“These [Danish employees] are people that we work for all the time. They should know this, right?”* “Not knowing the time” was often a problem for Filipino workers, forcing them to stay late or turn in work a day late due to delay in email responses. One Danish employee said he would sometime mitigate the effects of the time difference by replying to emails before he went to work in the morning. However, such actions were not typical for Danish workers. Given that the employees were highly professional in their training, expertise, and self-identity, “not knowing the time” had no positive interpretation for workers in the Philippines.

3.2.2 Requirement specifications

Requirements specification documents were a significant instigator of acts of closure. Filipino workers often did not understand the documents. Danish employees said the problems were cultural, indicating that they expected shared general knowledge, while Filipino employees thought the documents were poorly written. In a presentation, a Danish manager shared his views on writing the documents, using the forceful word “barrier”:

[I] think that our biggest challenge is specifying well enough...since they [Filipino workers] typically say that we have not specified well enough. “Oh but I believe we have.” “Well, we do not think so,” and then we dance around the issue. But it is also a question about...I believe there is a cultural barrier here.

Danish workers linked the challenges of requirements documents to lack of a shared culture. Culture thus became a barrier for dealing with the challenge. We asked why it was so difficult to understand the requirements documents. A Filipino manager said:

When we do these things [projects] it is shocking how many times we start off with a network diagram. You know, there is a database, there is a business layer, and there is a presentation layer, and there is all these things. What does it do? I mean just that, what is the context? I think that is the hard part...is getting them [Danish employees] to take that kind of fresh look at it and say what is this all about.

The Filipino manager asserted a problem of “context” meaning that domain knowledge was often underspecified, or altogether absent. Projects were introduced with abstract presentations of technical requirements depicted as network diagrams, databases, and so on. But there was often little information about the purpose of the system and the domain. At a workshop in the Philippines, one manager said: *“They [Danish workers] assume that we have the domain knowledge when we don’t.”* Another manager replied: *“Actually, I don’t think they see it as domain knowledge. I think they see it as common sense and...common sense is not so common. Common sense to one person is not common*

sense to another person.” Filipino employees needed more exposition of the domain and its properties to understand the nature of the system for which they were writing software.

While observing in the Philippine office, one developer asked us if we knew the meaning of a “P-number,” something he saw on a workflow sketch. Understanding the P-number requires knowledge of Danish taxation and government requirements. It is not common knowledge, and we had to look it up ourselves. We found a public Danish webpage on taxation (in Danish of course), and we were able to translate the meaning of the term. The Filipino employee could then continue his work. The researchers later learned that concepts such as P-number had been explained on more than one occasion both through presentations and documents. The episode with the P-number shows that both Danish and Filipino employees agreed that domain knowledge can be difficult to convey across national boundaries. The employees in Denmark had tried to convey the taxation concept, and their sentiment was that cultural differences must be the explanation for continued misunderstandings. Filipino employees opposed the cultural explanation by saying that the problem was that the way things were communicated hindered shared understanding. When Danes invoked a cultural explanation it stood out as a rhetorical move that foreclosed negotiation on how to meet the challenges related to domain knowledge and “common sense”.

3.2.3 Communication practices

We observed how employees in Denmark enacted closure through cultural interpretations and language regarding simple matters of everyday communication. For instance, Filipino employees preferred to address potential concerns in email, rather than through “outspoken” discussion. A Filipino employee we interviewed said:

When they [Danish employees] feel the need to speak they will just say it, and with us, we would tend to sort of think first: “Is it polite to say this?”...Sometimes if we are in doubt, instead of just asking directly, we would reserve it and probably just send it by an email.

Cultural patterns of action are invoked in this statement, but in more situated, nuanced terms than the Danish employees’ universalist assertions of cultural difference. The Filipino worker says, “*we would tend to sort of...*” using hedging language that connoted less definite tendencies as opposed to immutable cultural qualities. The worker explains the issue as a matter of preferred communication practice. It is not that Filipinos “do not reveal themselves,” but that they choose a different medium of communication. This discourse stands in contrast to the Danish manager we quoted above who said, “But it is exactly things like language.” He identified language as problematic, but at the same time kept the door open to other, unspecified aspects of culture (“things like”) as sources of difference between workers. The Filipino worker’s explanation of the problem regarding media choice preserves the possibility of negotiation and change. With some discussion, the email might turn into a Skype call or a conversation with an on-site manager, or some course of action moving toward problem resolution. It is more difficult to negotiate collaborative solutions when difference based on national origin—something an employee cannot change—is constructed as the root of the problem.

While Danish employees often generically mentioned the “language barrier” as a cultural problem, Filipino employees expressed concerns about language pursuant to the fact that they often received emails and documents written in Danish. During a workshop in the Philippine office a manager said: “*We had that email that was sent to [the whole department], all in Danish. And we had to reply: ‘Oh, please write in English, because you have Filipino counterparts.’*” Sometimes Filipino workers could manage the Danish by using tools such as Google Translate, but it was far from an optimal solution. We observed the translation issues when we shadowed Filipino developers. Many had Danish vocabulary written on the wall at their workstations, and they knew the keyboard shortcuts to specific Danish letters. A few were even able to translate words and sentences in Danish because they had grown so used to the use of Danish. The workflow sketches, for example, were all in Danish, and, as Danish observers, we were often asked to translate passages in the sketches. Filipino employees had clearly tried to adapt to some of the translation issues, but it was obviously difficult to understand the

full context of the Danish language. Filipino workers experienced language as a practical problem for the collaborative work, but they did not construe it as a problem of culture.

3.2.4 Closure mediated through technology

We observed how employees in Denmark used collaborative tools in enacting closure. The collaboration suffered from problems in the use of mediated communication such as email and instant messaging. For example a Filipino employee said in an interview, *“Whenever we sent them emails it was dead silence.”* Filipino employees were often dependent on getting answers to questions or status confirmations from Danish colleagues before they could move ahead with their work. Lack of response left them stranded. They felt it was generally difficult to get hold of Danish colleagues. A Filipino worker said that Danish workers seemed to be constantly offline: *“I guess that this is just the hard part wherein you really can’t get anything from them because they are out of reach or [you get] no replies from emails.”* This feeling was echoed by other Filipino employees who generally applied a more fluid approach to mediated communication tools. They were constantly engaged in online conversations and sometimes used instant messaging with people sitting right next to them! In interviews these workers explained that they preferred mediated communication because it was less intrusive. Their colleagues could keep working without being interrupted. Conversations could be archived for later use. By contrast, Danish employees were often marked “offline” in Microsoft Communicator (their instant messaging system), thereby delaying communication. When we asked a Danish manager in an interview about response time problems he said: *“Well, the issue about us [Danish employees] responding late is just the tip of the iceberg, right? What is really behind these things? They [Filipino employees] do not tell us...because it is a very different culture.”* Here we have a cultural iceberg upon which Filipino employees’ concerns about response times are taken to be a small signal of a larger problem stemming from different cultures. From the Filipino workers’ point of view, lack of availability in one of the key official channels of communication foreclosed open negotiation of problems. As one employee in the Philippines said:

Yeah, so it is quite hard...and there is Communicator all the time, and that helps as well, but for this project some of the key persons that needed to be asked were always away. I guess they are with a client or something.

Being “always offline” is charitably explained as “being with a client or something.” This formulation offers the benefit of the doubt, although given the professional context, the Filipino worker was well aware that employees in Denmark do not constantly have client meetings. Being left with “dead silence” provoked anxiety that someone might be angry: *“I have sent you the email and then there would be dead silence, and I would not know if he was mad at me because I am asking all these questions.”*

Acts of closure were much less apparent during video meetings. First, members were present in the meetings, so they could communicate directly with one another. Unlike email and IM, which often produced “dead silence,” members from both offices attended regular, weekly video meetings. There was a meeting for testers and one addressing technical issues. The test meetings usually consisted of one Danish tester, 1-2 Filipino testers, and one Filipino systems analyst. These meetings were not only task-focused but also helped establish common ground. The technical meetings had a Danish IT-architect and 1-5 Filipino developers. Those attending the testing and technical meetings came to share understanding of the complexities of the work, and the dependencies between pieces of the work done in both locations. Culture was never mentioned as a problem (although it might be brought up in a playful manner and sometimes included humorous exchanges). It is easy to ignore email and instant messages because there are so many of them, and a technology such as IM can be turned off, but video conferencing requires the commitment of scheduling and a mediated version of face-to-face interaction. Workers from both offices attended the meetings. The relative difficulty of enacting closure during videoconferences became an asset for collaborative problem solving in the cross-national team.

3.2.5 Closure as being out of the loop

What is striking about the incidents of closure we have presented was their one-sided character. The effects of closure were felt only in the Filipino office. Of course most employees there were Filipino, but there were also some Danish employees who had spent time in the Philippines. Danish employees who had worked in the Philippine office for longer than three months came to agree with many of the things Filipino workers said, including the feeling of being “out of the loop”. If challenges such as time difference and response time were truly cultural issues, the researchers would not likely have heard the following from Danish employees in the Philippines: *“I too lost something, so it is not cultural. It is purely about the distance that makes a difference.”* This employee had had a long tenure with GlobalSoft, yet he felt the impact of distance during the time he spent in the Philippines. We spoke to other Danish employees who had worked in the Philippine office and they shared similar stories. A Filipino manager said, *“You are out of the loop, you are far away, you get forgotten.”* This finding is consistent with the literature on problems of distance (Olson and Olson 2000, Cramton 2001, Hinds and Bailey 2003, Hinds and Mortensen 2005).

3.3 Understanding motives for closure

Why would employees in Denmark practice exclusionary acts of closure? It is a difficult task to identify underlying motives behind actions in a global collaboration because of the complexity of the activity. However, we point to conditions that may suggest why we observed a consistent pattern in which only Danish workers practiced closure. These conditions include uncertainty regarding the future, anxieties about job security, resistance to change, and questioning the efficiency and economic value of offshoring.

3.3.1 Developing a global mindset

The Danish employees had a history of working independently, which they very much enjoyed. In an interview, a Danish worker described the situation before the Filipino workers joined the company:

This department is established on the basis that we develop everything by ourselves. We do not have a good history of establishing partnerships. If we want a security component and we know that a company in town makes it, then we would still prefer to make it ourselves. Because this way we can maintain it if it does not work. Which is why we have a long history of developing everything by ourselves. We feel better that way. That is the culture in this department.

A preference for working independently was validated, in the eyes of the Danish workers, when the Filipino workers first joined GlobalSoft and continued using fixed-priced contracts, a practice from their supplier days. Danish employees felt that this arrangement did not commit Filipino workers to full responsibility for project outcomes, and that Filipino employees were mostly concerned with covering their estimated hours. Several Danish employees we interviewed suspected Filipino employees of padding time estimates. Suspicion and resentment over perceived responsibility created an “us-them” situation. A Danish employee said in an interview, *“We act a lot like ‘them and us,’ in my experience.”*

However, the contract was changed during the second year, in 2010. All employees were made equally responsible for project outcomes. Just as we began observations, a key project that affected several GlobalSoft departments was launched. Initially it was to be developed solely by Filipino workers. But shortly after the project started, the Danish employees who were involved left the company. The new people assigned to the project did not believe the Filipino employees would be able to deal with complexity of the work, and they reassigned 50% of the work back to Denmark. This reassignment was made despite increased costs and delays. Incidents such as the task reassignment, and suspicions about budgetary practices, indicate the mistrust Danish workers felt, and their troubled sense that established patterns of work in which they had long been comfortable were being disrupted by new global arrangements.

Filipino workers, by contrast, had “always been global.” They started with GlobalSoft when it was an outside supplier, and the work spanned national boundaries from the beginning. Understanding how to manage cross-national work was part of being a professional. A Filipino manager spoke in an interview about professionalism: “[P]eople hav[e] to get used to being more professional, and people hav[e] to realize...that there's a global component in the work that we're doing.” A global-inflected sense of professionalism was evident in patterns of social media use. Filipino employees were more or less constantly online, typically conducting several simultaneous instant messaging conversations. In Denmark, employees often marked themselves “offline” by default in Microsoft Communicator, making it difficult for Filipino colleagues to approach them. Danish workers would walk to each other's desks and have short discussions, or talk across the tables in open office space. While the Danish employees were somewhat older (suggesting a possible generational effect), they were sophisticated technical workers and could have easily chosen to use mediated communication tools more often had they wished to.

Comfort speaking English interacted with media choices. Filipino workers were accustomed to English. Their conversations moved fluidly between Tagalog (the local language) and English. Being an American colony for 50 years had resulted in a high proficiency and familiarity with English in the general Philippine population. Filipino workers thus did not mind picking up the phone. Though English is the official work language at GlobalSoft, and Danish employees were proficient in English, they spoke Danish at the office, choosing English only to talk to Filipino colleagues. They were less comfortable picking up the phone than Filipino workers. At a workshop in the Philippines, a Filipino manager said: “Some people [Danish employees] will not take the call. We had experiences when we were asking to just take a face call [a video call] and they refused. They would rather do the communication by email or chat.” When we asked the manager why Danish employees refused calls, she said, “They were concerned that their English was not good enough, so that was the issue.” Danish employees were not completely unaware of their tendency to stick to what they were used to and the resultant problems. During an interview, a Danish manager said:

But if we [Danish employees] were more willing to accept this new model in Denmark, if people could convince themselves that this is exciting...instead of seeing it as something threatening. But we are so strained by the way we do things in Denmark.

Questions about job security (“there is no job guarantee”), Danish workers' future roles in the company, disruptions of historical work patterns, and the less practiced use of English, were critical factors that influenced interactions with Filipino employees. Instead of embracing the collaboration and a global orientation, Danish employees resisted change by invoking culture as “a barrier”. Closure arose from tensions surrounding a desire to maintain valued historical ways of working.

3.3.2 Uncertainty about the future

The transition to working with offshore colleagues created increasing uncertainty and pressure on Danish employees. A Danish manager said, “There is no doubt that people are scared of losing their jobs, especially during these times where you cannot find anything else, right?” A Danish manager told us that even the management group did not know exactly which roles would be needed in the near future. In an interview he said, “A few people have dared to ask the question: ‘How does this affect my job? And what will my future look like?’ And these questions are still hard to answer.” The Danish manager admitted that the role and future development of the Philippine office were unclear even to the management group. He said, “[W]e have not yet reached a structured process where we know exactly [what will happen]...It grows organically, how much we use the Filipinos.” Offshoring was undertaken by GlobalSoft to lower perceived project costs. Lower bids could be prepared, which could potentially create or maintain jobs. A Danish manager told us during an interview, “In reality, the people most dedicated to offshoring and the advantages of it, are the top management group.” The perceived advantages of offshoring were cost-reduction, scalability, and competitiveness. But offshoring also increased the need for coordination and communication. Many Danish workers found

it difficult to see the benefits of offshoring, observing that project management often became more complex. In an interview, a Danish manager described how an offshore project “[becomes more] stringent and requires more documentation and coordination, and has a larger risk than having your own team.” The risks of delays and cost overruns concerned project leaders who dealt with the day-to-day running of the projects.

Danish employees outside top management experienced the extra coordination as a burden, and expected problems when working with Filipino employees. A Danish employee said in an interview, “*The Filipinos have delivered as expected with the usual extra iterations, which makes the project more costly and therefore more costly for us.*” He referred bluntly to the “usual” extra work expected when Filipino workers were involved, which he attributed to added iterations arising from mistakes and misunderstandings. The additional workload increased costs and sometimes delayed projects. Many Danish employees we spoke to shared these views. There was considerable skepticism regarding the actual economic benefits of offshoring. Zimmerman reported similar findings in her study of global software development groups of German and Indian developers, observing that negative attitudes towards an offshoring partner could result in reluctance to transfer “non-routine, complex tasks, in order to reduce quality issues and additional workload” (2012).

We do not know of any economic analysis that brings delays and failures into calculating true costs at GlobalSoft. Danish employees worried about the “invisible hours” in which highly paid Danish IT-architects spent much time correcting minor misunderstandings instead of using their competences more effectively. Some Danish employees did not believe there had been a single offshoring success story. One said in an interview, “*There has not been a success story yet. If they existed they would have been talked about. I am certain of that.*” Another told us, “*The history you hear in hallways is that what they [Filipino workers] deliver is scaled inadequately and performance is generally poor.*” Skepticism and doubt about the adequacy of offshoring put Danish workers in a defensive position in which acts of closure were expressions of concern over job uncertainty and frustrations regarding the efficacy of altering work patterns that had served well in the past.

4 Discussion

We have described how Danish employees used the rhetoric of culture as an act of closure. Strategies reported in other studies, such as lack of interaction and criticism, are very direct, and convey pointed messages, whereas the rhetorical acts of closure we discovered were more subtle. In Metiu’s case, the Indian workers she studied became so frustrated as targets of closure, they left the US-based project they had been working on. In our case, Danish employees’ acts of closure were aimed at maintaining tasks and traditional work practices, rather than completely shutting out the Filipino employees. We also observed more typical acts of closure at GlobalSoft, but they seemed less pervasive than what Metiu reported. But they did occur. For example, at one point during a video meeting, we observed workers in the Philippines who were surprised to learn that they would continue their work with a new version of a product.

Filipino employee (with disbelief): There is a new version of [the product]?

Danish worker: Yes, there will be one in the future.

Second Filipino employee: We did actually not know that we could expect a new version of [the product]. I have to pull some strings to get the same developers back to [the project].

The Philippine office was yet again being left out of loop regarding essential information. Metiu observed that such actions are exacerbated by historical and economic status differences, as well as geographic distance which “lowers the costs of exclusion” (Metiu 2006). In a face-to-face interaction, an act that would be grounds for embarrassment (such as not answering a question posed by a colleague, failing to read a document sent several days previously, or withholding information about project scheduling) becomes less problematic in a mediated environment.

Our research shows an asymmetric relation in acts of closure centered in notions of culture. Danish and Filipino workers in the same department at GlobalSoft behaved differently. By contrast, other studies point to reciprocal constructions of cultural difference. Marrewijk found that Dutch and Indian employees simultaneously constructed cultural differences to strategically improve their relative power in their organizations (Marrewijk 2010). Indian workers cultivated an image of cultural adeptness at avoiding conflict and disharmony to legitimize the idea that they should have more contact with clients. Dutch employees portrayed themselves as more punctual, indicating that they should remain in control of planning and leading projects (Marrewijk 2010). In our case, the rhetoric of culture had implications for the use of technology. For instance, Filipino workers engaged in instant messaging conversations with co-workers located right next to them, and preferred IM because conversations could be archived and interruptions reduced. IM conversations between Denmark and the Philippines happened much less. Danish employees were less inclined to choose synchronous modes of mediated communication with Filipino employees because of unease with English, but also due to a presumed cultural “barrier” in which they saw not sharing the “same logical background” as another impediment to understanding in mediated conversation. Acts of closure were easier to execute with the use of asynchronous communication tools such as the requirements specification documents, email, and instant messaging. However, videoconferencing seemed to work in the opposite direction because it required participants to explicitly commit to the collaboration. Media choice can thus help managers contend with closure although of course it does not solve all the problems. More importantly, understanding broad processes such as closure can aid managers in interpreting employees’ media choices and addressing root causes of collaboration problems. Our contribution adds to the growing body of literature that critically investigates cultural explanations in offshoring work (Marrewijk 2010, Boden 2012). Each of these studies offers similar but slightly different problems with cultural explanations. We propose that in construing national culture as a stable, persistent condition, options for negotiation and discussion which might have brought the Danish and Filipino employees into a state of more equitable relations, were foreclosed, and static relations tended to persist. We suggest that teams involved in global software development should openly acknowledge the risk of closure strategies to strengthen the use of synchronous mediated communication such as video conferences, which promoted collaboration through the necessity of commitment.

Offshoring software development tasks is a key means by which today’s corporations strive to stay competitive. Offshoring will continue to increase in the future, and internal competition over assignments in cross-national groups is likely right from the beginning of such projects. As one Filipino employee said: “*You get that feeling that everyone is competing to get that next big project.*” Success and failure were important themes in Globalsoft discourse. Marrewijk argues that in global software teams, one group may actually benefit from failures in the collaboration (Marrewijk 2010). A “paradox of success” arises in which successful offshore projects eventually shift the balance of power toward the offshoring group, leaving the onshore group at a disadvantage. Successful projects will instigate more offshoring because of the logic of cost reduction. But this shift is a loss for the onshore group, potentially leading to the relocation of work and skilled jobs to offshore locations. Failures of collaboration in which the offshore employees are blamed for problems may thus benefit onshore employees. The Danish managers in our study said that the growth of the offshore group was “organic,” suggesting that, as Marrewijk argued, successful offshore projects eventually beget more offshore projects. Danish employees worried about what this trend would mean for their future.

Danish employees shared their uncertainties about future employment and fear of losing control. Such problems have been documented in other studies of global software development (see Oza and Hall 2005). In a sense, offshoring is something of a “zero-sum game,” in which one group gains at the expense of the other. Zimmermann found that German developers were “seen to actively seek evidence for Indian mistakes in order to argue against the transfer [of tasks]” (2012). In our study, gains for Danish employees would similarly result from perceived lower quality offshore work since more work would then be conducted in Denmark. This was precisely what happened when Danish

employees pulled back 50% of a project intended to go the Philippines, putatively because they could not rely on the Philippine office to do the job correctly. The long-term gains of remaining competitive in the market may not hold much sway with employees who must deal with day-to-day uncertainties, frustrations, and fears. Short-term benefits legitimize closure to preserve opportunities and resources for employees in the Danish office (see also Metiu 2006).

Danish workers' discourse indicated that culture is stable, habitual, historical, ordinary, primal. They described themselves as "Scandinavian," suggesting a historical cultural realm even beyond Denmark. They spoke of culture in powerful exclusionary metaphors such as "borders" and "barriers." Filipino workers were also aware of culture though they did not construct notions of culture to exclude Danish employees. Enacting closure would have worked against their goals of solving daily problems which enabled them to be more valued global workers. Describing difficulties as tractable problems such as document production, rather than problems caused by an immutable characteristic, positioned them to actively explore solutions. Consonant with the logic of flexible problem solution, we observed that Filipino workers used culture as a relatively flexible construct in which variation was possible, and in which a person could even go against prevailing cultural norms and choose the cultural traits he or she wanted. In an interview, a Filipino manager who had just described her culture as "polite," then excluded herself from this categorization, saying: "*So, [the norm] comes out as being polite, or as not being able to say no. I'm probably more tactless. (Laughs.)*" The manager acknowledged the cultural norm in the Philippines, but at the same time, said that it did not apply to her. She was "tactless," and this did not concern her. The manager continued, saying:

We're very hierarchical. You will see some people who still call me Ma'am or Miss. I mean, I call [the general manager by his name]. Not like, I'm not about to call him sir. But sometimes I tease him and call him boss. But, you know, that's more like a joke.

An individual could play with cultural norms, even joking with a superior, in a culture of hierarchy. We noticed that Filipino workers told stories about playing around with culture while Danish workers did not. For example, a Filipino employee remarked in an interview: "*For me, I don't see there is a problem. Maybe because I am less offended, so I guess it depends on the person.*" The nuanced way in which Filipino employees addressed the collaborative challenges at GlobalSoft (such as becoming "tactless" to be a better manager or "less offended" to be a good worker), seems to underscore the "professionalism in a global context" Filipino workers valued, in contrast to Danish workers who ascribed the challenges of collaboration to static cultural differences.

Identifying "cross-national" problems focuses the analysis more precisely on the particular cultures and organizations under study. The term "global," while useful, is less precise, and effaces particular historical and sociotechnical conditions which should be taken into account in understanding closure. Socio-political motives such as resistance to change impeded use of communication technologies for purposes of cross-national collaboration not because of faults in the design of the technologies, but because of a pattern of acts of closure (see also Meissonier and House, 2010). Danish workers often invoked culture as the root of problems of collaboration, but when questioned in interviews, they sometimes gave vague answers about what they meant, or retrenched to culture as merely language. Sometimes they were firm about specific cultural differences such as one employee who said: "*[Filipino workers] share a cultural characteristic where you do what you are told and do not ask questions unlike how we do it here in Denmark.*" Ybema and Byun observed: "Within the politicized context of a multinational corporation, organizational actors may play up or play down, praise or dispraise, or even ignore or invent culture and cultural differences." (Ybema and Byun 2009). The asymmetry of the uses of culture at GlobalSoft exemplifies Ybema and Byun's characterization of the flexible dynamics of cross-national collaboration. Asymmetry in closure may appear when national groups working together have differing interests and concerns. Marrewijk commented that "[Managers] should learn to understand how their own behavior is interpreted by the 'other', given (a)symmetric power relations" (Marrewijk 2010). In our study, we saw only Danish employees

construct a cultural “other,” while Filipino employees spoke of problems of coordination, communication, and professionalism in their efforts to manage collaboration.

5 Conclusion

In our study we found that employees in Denmark invoked the rhetoric of culture in an attempt to maintain power and influence with respect to their work practices, while Filipino employees resisted cultural explanations, providing other causal explanations. Closure occurred in response to new conditions of work and uncertainties and concerns generated by offshoring. In construing national culture as a stable, persistent condition, options for negotiation and discussion which might have brought the Danish and Filipino employees into a state of more equitable relations, were foreclosed, and static relations tended to persist. Closure affected media choices and patterns of everyday communication. In particular, we observed how modes of asynchronous mediated communication enabled acts of closure through the rhetoric of culture, whereas video conferences seemed to promote collaboration through the necessity of commitment, and possibly through the face to face nature of videoconferencing, a topic that requires further study. Teams involved in GSD should openly acknowledge the risk of closure strategies. It would be useful to establish a critical approach to cultural explanations and question them whenever possible. The rhetorical use of closure is disabling for finding constructive solutions to challenges, and practitioners should strive to look beyond “typical” cultural explanations.

Acknowledgements

This research has been funded by the Danish Agency for Science, Technology and Innovation under the project "Next Generation Technology for Global Software Development", #10-092313. We would like to thank our study participants who generously spent time with us and shared their thoughts. We are grateful to anonymous reviewers for a close reading of the manuscript and their helpful suggestions.

References

- Bidwell, M. J. (2013). What Happened to Long-Term Employment? The Role of Worker Power and Environmental Turbulence in Explaining Declines in Worker Tenure, *Organization Science*, 24(4).
- Boden, A., Avram, G., Bannon, L. and Wulf, V. (2012). Knowledge Sharing Practices and the Impact of Cultural Factors: Reflections on Two Case Studies of Offshoring in SME, *Journal of Software Evolution and Process*, 24(2), 139 - 152.
- Bottero, W. (2005). *Stratification: Social division and inequality*, London and New York: Routledge.
- Carmel, E. and Tjia, P. (2005). *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*, United Kingdom: Cambridge University Press.
- Chand, D., David, G., Galliers, R. and Kumar, S. (2009). An Investigation of How Culture Impacts Global Work: Unpacking the Layers of Culture, in *European Conference on Information Systems (ECIS)*, 537 - 555.
- Conchúir, E. Ó., Ågerfalk, P. J., Olsson, H. H. and Fitzgerald, B. (2009). Global Software Development: Where Are the Benefits?, *Communications of the ACM*, 52(8), 127-131.
- Cramton, C. D. (2001). The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration, *Organization Science*, 12(3), 346 - 371.
- Cramton, C. D. and Hinds, P. (2005). Subgroup Dynamics Internationally Distributed Teams. Ethnocentrism or Cross-National Learning?, *Research in Organizational Behavior*, 26, 233 - 265.

- Deshpande, S., Richardson, L., Casey, V. and Beecham, S. (2010). Culture in Global Software Development - A Weakness or Strength?, in International Conference on Global Software Engineering (ICGSE), Princeton, New Jersey, IEEE, 67 - 76.
- Diamant, E. I., Fussell, S. R. and Lo, F.-I. (2008). Where Did We Turn Wrong?: Unpacking the Effect of Culture and Technology on Attributions of Team Performance, in Conference on Computer Supported Cooperative Work (CSCW), San Diego, California, USA, ACM Press, 383 - 392.
- Ehrlich, K. and Cataldo, M. (2012). All-for-One and Ane-for-All?: A Multi-level Analysis of Communication Patterns and Individual Performance in Geographically Distributed Software Development, in Conference on Computer Supported Cooperative Work (CSCW), Seattle, Washington, USA, ACM Press, 945 - 954.
- Eisenhardt, K. M. (1989). Building Theories From Case Study Research, *The Academy of Management Review*, 14(4), 532 - 550.
- Gregory, R. (2010). Review of the Offshoring Literature: The role of Cross-Cultural Differences and Management Practices, in European Conference on Information Systems (ECIS), Paper 147.
- Hinds, P. J. and Bailey, D. E. (2003). Out of Sight, Out of Sync: Understanding Conflict in Distributed Teams, *Organization Science*, 14(6), 615 - 632.
- Hinds, P. J. and Mortensen, M. (2005). Understanding Conflict in Geographically Distributed Teams: An Empirical Investigation, *Organization Science*, 16(3), 290 - 307.
- Jarvenpaa, S. and Leidner, D. (1999). Communication and Trust in Global Virtual Teams, *Organization Science*, 10(6), 791 - 815.
- Jarvenpaa, S. L., Knoll, K. and Leidner, D. (1998). Is Anybody Out There? Antecedents of Trust in Global Virtual Teams, *Journal of Management Information Systems*, 14(4), 29 - 64.
- Kayan, S., Fussell, S. R. and Setlock, L. D. (2006). Cultural Differences in the Use of Instant Messaging in Asia and North America, in Conference on Computer Supported Cooperative Work (CSCW), Banff, Alberta, Canada, ACM Press, 525 - 528.
- King, W. R. and Torkzadeh, G. (2008). Information Systems Offshoring: Research Status and Issues, *MIS Quarterly*, 32(2), 205 - 225.
- Krishna, S., Sahay, S. and Walsham, G. (2004). Communications of the ACM', International Conference on Intercultural Collaboration, 47(4), 62 - 66.
- Kumar, R. and Jha, G. N. (2010). Translating Politeness Across Cultures: Case of Hindi and English, in International Conference on Intercultural Collaboration, Copenhagen, Denmark, 175 - 178.
- Mahadevan, J. (2011). Power/Knowledge in Postcolonial Settings: The Case of IT Bangalore, *Interculture Journal*, 10(13), 61 - 81.
- Malone, T. W. (2004). *The Future of Work: How the New Order of Business Will Shape Your Organization, Your Management Style and Your Life*, Boston, Massachusetts: Harvard Business School Press.
- Marrewijk, A. v. (2010). Situational Construction of Dutch-Indian Cultural Differences in Global IT Projects, *Scandinavian Journal of Management*, 26(4), 368 - 380.
- Metiu, A. (2006). Owning the Code: Status Closure in Distributed Groups', *Organization Science*, 17(4), 418 - 435.
- Noll, J., Beecham, S. and Richardson, I. (2010) 'Global Software Development and Collaboration: Barriers and Solutions', *ACM Inroads*, 66 - 78.
- Olson, G. M. and Olson, J. S. (2000) 'Distance Matters', *Human-Computer Interaction*, 15, 139 - 178.
- Oza, N. V. and Hall, T. (2005) 'Difficulties in Managing Offshore Software Outsourcing Relationships: An Empirical Analysis of 18 High Maturity Indian Software Companies', *Journal of Information Technology Case & Application Research*, 7(3).
- Ravishankar, M. N., Pan, S. I. and Meyers, M. D. (2013) 'Information Technology Offshoring in India: A Postcolonial Perspective', *European Journal of Information Systems*, 22, 387 - 402.
- Søderberg, A.-M. and Holden, N. (2002) 'Rethinking Cross Cultural Management in a Globalizing Business World', *International Journal of Cross Cultural Management*, 2(1), 103 - 121.

- Strauss, A. and Glaser, B. G. (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research*, United States: Rutgers.
- Tang, J. C., Marlow, J., Hoff, A., Roseway, A., Inkpen, K., Zhao, C. and Cao, X. (2012) 'Time Travel Proxy: Using Lightweight Video Recordings to Create Asynchronous Interactive Meetings', in *Conference on Computer-Human Interaction (CHI)*, ACM Press, 3111 - 3120.
- Weber, M. (1978) *Economy and Society*, Berkeley, CA: University of California Press.
- Weeden, K. A. (2002) 'Why Do Some Occupations Pay More than Others? Social Closure and Earnings Inequality in the United States.', *American Journal of Sociology*, 108(1), 55 - 101.
- Ybema, S. and Byun, H. (2009) 'Cultivating Cultural Differences in Asymmetric Power Relations', *International Journal of Cross Cultural Management*, 9(3), 339 - 358.
- Zimmermann, A. (2012) 'Offshoring Attitudes and Relational Behaviours in German-Indian Offshoring Collaborations: Reflections from a Field Study', in *International Conference on Intercultural Collaboration (ICIC)*, Bangalore, India, 107 - 118.

The next paper to be presented is paper 3

**Jensen, R. E., & Bjørn, P. (2014). Demonstrating Commitment in Practice:
Tracing & Gaining Attention. *Submitted to Journal of New Technology, Work
and Employment.***

Demonstrating Commitment in Practice: Tracing & Gaining Attention

Rasmus Eskild Jensen and Pernille Bjørn

IT University of Copenhagen, Technologies in Practice, Rued Langgaards Vej 7, 2300
Copenhagen S, Denmark, email: [raej, pbra]@itu.dk

Abstract. In this paper, we investigate commitment as a practice – a doing – as an important feature of collaborative practice. We explore the ways in which individual practices at work demonstrate commitment as part of a collaborative engagement, and how these work practices becomes part of the collaborative work arrangement. Reporting from an ethnographic study that was conducted over the course of three years in a global software development setup between Denmark and the Philippines, we found that the practices that demonstrate commitment are centered around tracing remote colleagues' attention through temporal and artefactual clues captured by technology mediation. We use collective attention to identify the work practices by which employees in a distributed collaboration gained and traced the direction of attention of other employees in the collaboration. Demonstrations of commitment are founded on attempts to establish collective attention among employees concerning the task, the process, and the people involved.

Keywords. Collective Attention, Temporal traces, Artefactual traces, Commitment, Ethnographic Case Study, Global Software Development .

1 Introduction

Research have argued that commitment in virtually distributed teams is critical to overcoming the challenges in distributed knowledge work (Bjørn et al. 2006; Jarvenpaa et al. 1999). Research on information technologies in the work place has become increasingly interested in the consequences on work–life boundaries of many workers and how this development does not necessarily “correspond to the confines of the traditional ‘workplace’” (Howcroft & Taylor 2014, p.2). One aspect of these consequences is related to the notion of commitment in distributed work places. In organizational behavior and human resource literature, commitment in the workplace is also viewed as strategically important due to the assumption that highly committed employees perform better (Meyer et al. 2012).

In the organizational literature, commitment has been defined in terms of employees’ engagement in a consistent line of activities (Söllner 1999), characterized by a persistent behavior over time where participants pursue the same goal regardless of the external diversity of the activities (Becker 1960). Previous studies on commitment focus on a behavioristic approach toward commitment based on the assumption that people are driven by commitment in ways that are rational, emotional (Scholl 1981), instrumental, or affective (Cohen 2007), and that we as researchers can have access to measure employees’ individual commitment. However recent research have come to question the link between employee commitment to the organization and the performance. For instance Cushen and Thompson (2012) found that the use of high commitment

policies in a knowledge intensive organization contradicted the everyday experiences shared between the workers. Interestingly, they also point to the fact the knowledge workers had a high performance while simultaneously being uncommitted (Cushen & Thompson 2012). If we believe that there is a link between commitment and performance but studies show that people can perform great without being committed then there might be a problem with the way we measure commitment. Using this discrepancy as a starting point we wanted to investigate an alternative view on commitment.

In our endeavor to re-think the fundamental approach to commitment, we turned to the *practice approach* (Law 2004; Reckwitz 2002; Schatzki et al. 2001). The practice approach is concerned with how people in practice accomplish their work. The ontological foundation for the practice approach specifies that the only practices we, as researchers, can access and report upon are manifestations of the connections holding together people, artifacts, and activities (Bjørn & Østerlund 2014; Bjørn 2011). Adopting the practice approach suggests that we need to re-think the concept of commitment because we do not have access to open up peoples' minds; instead we have to investigate how commitment is enacted in organizational practices. This theoretical turn to practice brings with it particular assumptions about our research on commitment. First, we see commitment as a collaborative activity rather than an individual orientation. Collaboration is the practices where multiple people are mutually engaged within a common field of work, and where individual activities changing the state of the common field of work have direct impact on the collaborative partners, who then have to act

accordantly (Schmidt et al. 1992). Thus, our approach to commitment is that it is an accomplishment of mutually interlinked practices between people, artefacts, and activities. Second, by viewing commitment as a collaborative practice, we submit to the notion that even though commitment is an entity that includes various invisible aspects, such as values, beliefs, and attitudes, we do not have access to studying such aspects. It is impossible to open up the minds of the people we study (Boden et al. 2009a; Kudaravalli et al. 2011). Instead, we study commitment by referring to its manifestations in the form of artefacts and practices within our case on global software development, which then becomes our center of attention.

Applying a practice approach to the previous literature on commitment, we then have to identify the possible manifestations we might examine in the empirical material. In real life, commitment might appear to be a confusing, irremediable “mix of interests, side bets, acts of commitment and consequent behavior” (Becker 1960). Given this elusive character of commitment, commitment may only become visible when discrepant events change a given situation and reconsideration of the collaboration becomes evident (Majchrzak et al. 2000). Commitment is thus part of the collaborative work practices, which people rarely articulate.

The contribution of this paper is two-fold. First, we identify two different strategies by which the IT developers in our study manage to demonstrate commitment, despite being geographically distributed, as the work practices of enacting temporal and artefactual traces within the technologies. Second, we

propose a conceptual understanding of commitment as a practice and feature of collaboration, which can further develop the basic foundations for how commitment is understood in the work place.

The paper is structured as follows. First, we construct a theoretical basis for how to understand commitment as a practice. Second, we present the research method, the empirical case, and the analytical approach applied on the case. Third, we present strategies for demonstrating commitment from the case through two types of tracing: temporal and artefactual. This is followed by a discussion of the empirical findings creating the theoretical foundations for how to conceptualize commitment as a practice in geographically distributed collaborations. Finally, we conclude.

2 Commitment

Traditionally, in the literature on collaboration and collaborative technologies, commitment has never held centre stage, but rather has been mentioned as a side note or aspect of the collaborative feature in focus. Investigating coordination as a feature of the design of collaborative systems commitment played an important part in projects such as The Coordinator (Winograd & Flores 1986) and the CHAOS project (Simone & Divitini 1999). In both these projects the systems' designs and architectures were based upon the language perspective, and commitment was thus 'designed' as an action and a functionality by which collaborative partners explicitly demonstrated their commitment to accepting a

task within the system. While the CHAOS project was designed to support coordination (Simone & Divitini 1999) and The Coordinator to support communication (Winograd & Flores 1986), the demonstration of commitment was the same – namely, in explicit agreements on tasks, which is negotiated (Bjorn & Hertzum 2006). When collaborators finally reach an agreement, commitment is established; thus, making it easier for collaborative partners to know what others are doing on their own to support the joint task (Dabbish et al. 2012).

Other work on commitment focuses less on the language action, and more on the common processes by which the collaborative partners agree on work (Søderberg et al. 2013) and develop conventions for conversation (Mark 2002a). Investigating the distributed administration of the German government in offices in Bonn and Berlin after the fall of the Berlin Wall, Mark (Mark 2002a) points to several aspects that made it difficult for employees to negotiate and commit to joint conventions of how to communicate using digital means. In particular, she points out how commitment to digital conventions requires specific awareness information on the social, behavioral, and environmental situation, since this becomes critical for articulating the conventions (Mark 2002a, p.359). By having access to others' actions we are better equipped to make sense of others' activities, including the ways in which they demonstrate commitment. As Robinson phrases it:

“It was important to know how committed other groups were to their claims – not just that they had made them. Conversely, it would have been difficult and confusing to know about the motive and commitment to a claim without knowing exactly what was being proposed. It was

important that precision and ambiguity were separated out, and that they could interact, each helping to define, and providing a reference point for the other.” (Robinson 1991, p.43)

What Robinson argues is that it is not simply enough to know that other people explicitly declare their commitment to complete a task. It is equally important to know the degree of commitment to reduce ambiguity. Thus, demonstrating commitment becomes both a process of explicitly declaring to do a task, but also to demonstrate through various types of awareness information the degree to which this declared commitment is actually going to prevail. So how do people determine the degree of commitment of others, and thus act accordingly? In a study of the role of personal profiles in virtual project teams, the degree of commitment of others was established by knowing about the other person’s profession (Rusman et al. 2013). For example, if a person was perceived to be building up a career and the success of the project would support this goal, other people would assume the person would spend significant time and energy on the project. This feature of commitment can also be explained in terms of side-bets. Side-bets is the individual practice through which people take an action that increases the cost associated with discontinuing other related actions and, thus, invest time and effort in specific actions – betting that these activities will pay off at a later time (Allen & Meyer 1990a). While the degree of commitment is decided based upon the likelihood of people actually fulfilling their commitments, the interesting aspect is how people manage to demonstrate their degree of commitment in practice. Expressing commitment concerns gaining each other’s attention, and the most powerful way to gain attention and express commitment is by bodily presence (Nardi 2005).

“Commitment is an engagement denoting scope for ongoing communication for projects of mutual interest. Just ‘showing up’ communicates that one is attending to the relations necessary for carrying out such projects (...) the expression of commitment is not about actually doing the work or engaging in the mutual project; it is simply about being there.” (Nardi 2005, p.125)

What is interesting in this quote is that demonstrating the degree of commitment is not attached to actually doing the task; instead, physical presence is disconnected to the actual task. Being physically present as practice of gaining attention is done by simply stepping into the picture – being physical and tangible, rather than a digital interaction. However, in the cases of global software development, the fundamental premise is that the IT developers will be geographically distributed most of the time, thus bodily presence as a strategy to demonstrate commitment cannot be applied on a regular basis.

From the literature, we have now established that demonstrating commitment concerns the declaration of commitment to a task, as well as showing the degree of commitment. Also, we found that commitment can be detached from the actual task and can be demonstrated simply through bodily presence. However, in most cases within a globally distributed setting, this strategy will not be applicable. Previous research all points towards the same conclusion: namely that the local, the tangible, and the attention, all of which comes from physical proximity, seem to be crucial factors in demonstrating commitment. However, how do people who do not have physical proximity manage to demonstrate commitment? What are the conditions that make it possible to express commitment in geographically distributed collaborative settings? What are the ways in which distributed IT

developers manage to gain each other's mutual attention making it possible to demonstrate commitment? These are the questions we seek to explore.

3 Methods

The work presented in this paper was done as part of a larger five-year strategic research project: [PROJECT NAME]. The [PROJECT NAME] research project investigates both small, medium, and large software companies with global reach in order to increase the understanding of cross-cultural software development and to design new technologies that will support the work practices of global software development. We conducted a workplace study investigating the ways in which globally distributed IT developers manage to gain each other's mutual attention. Ethnographic workplace studies enable researchers to identify 'invisible' work practices and efforts that are not accounted for in verbal accounts (Luff et al. 2000; Blomberg & Karasti 2013). The empirical work reported on in this paper is based on one sub-project dedicated to study work practices within a medium-sized software company named GlobalSoft (a pseudonym). In the GlobalSoft study, we followed a project (the DAFIC project) that involved developers located in Denmark and in the Philippines. During the three-year study (December 2010–November 2013), we followed ethnographically (Forsythe 1999) the transformation of GlobalSoft from a Danish company working with outsourcing to the Philippines, to its becoming a global company with offshore units through its acquisition of the Philippine company.

Empirical context

GlobalSoft is of Danish origin, with its main office in Copenhagen, Denmark. However, since 2006, GlobalSoft has expanded its operations to include the Philippines, China, and various European countries. We focused on the collaboration between Denmark and the Philippines because this is where the majority of the software development is conducted. Over the course of three years the DAFIC project was an important GlobalSoft project that aimed to create a new software solution for the Danish public sector by establishing a new access infrastructure to the public service databases. The DAFIC system allows citizens and companies to gain access to all public websites with only one set of login credentials, making it easier to work across multiple public platforms. The project had a high level of technical and organizational complexity since the DAFIC system had to be technically connected to a large range of electronic systems already in use. In addition, the project got off to a really bad start because of an overly optimistic contract design. Despite these factors and delays, the project is now completed and implemented in Denmark. And, despite the problems, the IT developers involved in the DAFIC project expressed that this project had been exemplary in how collaboration and communication could be handled across the two locations in Denmark and Philippines. Trying to understand what kept the global collaboration from falling apart led us to initiate an analytical process where we systematically went through the empirical data looking for evidence of how practitioners managed to successfully collaborate despite all the problems. Interestingly, we found that the ways in which commitment was demonstrated

across the locations was an essential component for explaining the good collaboration.

Data sources

The research was initiated in December 2010 and executed by three researchers and one research assistant over the entire course of the three years. Critical events to our research included multiple on-site observations in Denmark and the Philippines, as well as a range of interviews and workshops we conducted over the years. In total, researchers conducted 28 semi-structured audio-recorded interviews (19 in Denmark, 9 in the Philippines) with an average length of 50 minutes. Employees at many different organizational levels were interviewed, from the corporate vice-president to junior developers (see Table 1). Observations were conducted in Manila, Philippines, for approximately four months over the course of four separate phases (December 2010, July 2011, November 2011, and January 2012), as well as in Copenhagen, Denmark, where the majority of the work included observations of the on-going DAFIC project over a period of 12 months. During this period in Denmark, the first author made observations for three to four days per week for five months, and then one to two days per month for the remaining seven months.

The observations and interviews focused mainly on understanding the cooperative work practices among globally distributed project members to identify challenges and opportunities in a global software development. During the data generation process, we continuously discussed our initial empirical

findings with the DAFIC employees, either during casual conversations or during official presentations and workshops. Four presentations during the fieldwork period (three in Denmark and one in the Philippines) communicated the findings to company representatives. Based on discussions, feedback from employees was noted and included in the different stages of analysis, and we used these feedback sessions to challenge and validate our findings. Similarly, four workshops (two in Denmark and two in Philippines) were conducted and video-recorded for further analysis. We also examined internal company documents, requirement specifications, and other official materials.

Table 1: Data Sources.

Time & location	Data sources	Use of data in analysis
November 2010, Denmark	14 preliminary audio-recorded and transcribed interviews with employees at GlobalSoft, ranging from developers, IT-architects and managers, to the vice president <ul style="list-style-type: none"> - Interviews lasting 30–60 minutes - Average length 50 minutes 	Establish understanding of the company and project structure in global software development Identify key challenges of the collaboration between Denmark and the Philippines
December 2010 – January 2011, Philippines	2 interviews with the manager and the project leader Observation of the everyday work practices Presentation of initial findings	Achieve initial understanding of the Philippine perspective
Spring 2011, Denmark and the Philippines	4 workshops conducted – 2 in Denmark and 2 in the Philippines	More detailed knowledge of the Danish and Philippine perspectives on the collaboration
July 2011, Philippines	3 interviews with team leader, tester, and manager 1 st workshop on opportunities and challenges in global software development 10 8-hour days of observations, including observing developers, project managers, and testers. Different types of meetings were observed.	Further understanding of the collaboration from the Philippine perspective
October – December 2011, Denmark	Observations of the DAFIC project 2–3 times a week lasting between 4 to 10 hours <ul style="list-style-type: none"> - Observing work practices 	Understand the everyday practice and challenges of global software development in DAFIC from a Danish

	<ul style="list-style-type: none"> - Observing team meetings, steering meetings, and client meetings - Informal audio-recorded interviews lasting 5–10 minutes 	perspective
<p>January 2012, Philippines</p>	<p>Observations of DAFIC four times a week lasting between 4 to 10 hours</p> <ul style="list-style-type: none"> - Observing work practices - Observing team meetings - Informal audio-recorded interviews lasting 5–10 minutes <p>Four audio-recorded interviews with manager, project leader, project manager, and system analyst</p> <ul style="list-style-type: none"> - Interviews lasting 30–60 minutes - Average length of 45 minutes <p>Audio recorded presentation of initial findings for management in the Philippines lasting 60 minutes including validating feedback</p>	<p>Understand the everyday practice and challenges of global software development in DAFIC from a Philippine perspective</p>
<p>March 2012 – October 2013, Denmark</p>	<p>5 audio recorded and transcribed follow-up interviews and observations in DAFIC (average length of 50 minutes.)</p> <ul style="list-style-type: none"> - Monthly follow-up observations - Attending evaluation meetings for DAFIC project - Questionnaire sent to the Philippine employees for follow-up <p>Presentation of findings for the team Presentation of high-level implications to the management group</p>	<p>Validating interpretation and understanding of the collected empirical data</p>

Data analysis

The data analysis emerged while we still were in the field, as well as through the various discussions we had with employees and the discussions we had amongst ourselves away from the field (Randall et al. 2007). The constant conundrum for us in these initial stages of analysis was that it was clear from both observations and interviews that the employees constantly reflected on and changed their ways

of collaborating while trying to figure out what worked best and what needed to be changed. However, it was clear to us that it was not because the collaboration across sites was increasingly problematic in terms of factors such as coordination or communication. The developers explained they believed that the collaboration had never been better across sites. Therefore, we analysed all the data material and, in particular, sought reasons for its seeming functionality (Harper 2000). What allowed the IT developer to experience beneficial collaboration despite all the problems? By scrutinizing the research material it became clear that there were, of course, situations during which communication and coordination had failed. However, we also found that one unique aspect of the collaboration was that the dispersed developers constantly directed their attention towards their remote colleagues in a reflective manner, as well as to the joint project itself. We began to play with the idea of commitment as a doing and as a practice, to determine whether the developers were managing to demonstrate commitment across sites in new ways, supporting the project's strong relationships between developers.

Through the literature on commitment, it became clear that ours was a unique case in which employees managed to demonstrate commitment through technology-mediation, and that this type of teamwork played a major role in terms of making the collaboration work. While the data from our empirical material is extensive, we have chosen to mainly focus on the development activity of testing.

4 Tracing and gaining attention

Our primary interest in this paper is to understand the ways in which IT developers involved in global software development demonstrate commitment in practice in a technology-mediated environment. Based on our analysis we found that demonstrating commitment is essentially a practice of employees paying attention to the task, the people involved, or processes required, and then getting the attention of collaborative partners in order to monitor how they are directing their attention – and then acting accordingly. We use collective attention to point to the practices at work by which employees, while collaborating, also follow where other employees are placing their attention and act accordingly. We propose that having mutual collective attention placed on a task is a demonstration of commitment in a collaborative setting, since it includes both an articulation of commitment combined with a demonstration of the degree of commitment.

Our specific interest in this paper is to identify the demonstration of commitment in a geographically distributed setting and, in particular, how technology plays a role in demonstrating commitment. Thus, our focus here is to identify and follow how digital traces from individual activity display commitment as part of the collaborative setup, and how attention placed on these traces aids practitioners in identifying the mutual collective commitment in a distributed testing environment. When we initially began our study of GlobalSoft in 2010, one of the first things we were told was that testing was an essential area

for understanding the collaborative work of global software development. The company's vice president at the time said: "*Testing is probably an area that you should investigate thoroughly because this is the phase during which the software is transferred back to Denmark.*" The testing stage marks when Danish and Philippine employees merge their efforts and are mutually dependent on each other. Testing practices were particularly interesting in the DAFIC project because IT developers coordinated these activities with the use of shared collaborative technologies, such as the test manager system, and shared a mutual responsibility for finishing the testing of products on time. Previous research has also found testing processes to be a critical aspect of software development (Rooksby et al. 2009; Martin et al. 2006), thus our focus here will be commitment as part of the collaborative testing practice.

In the following sub-sections, we first describe a typical test setup and show how testing was performed collaboratively in our case. Next, we will introduce how the testers used technology traces to inform their work and identify two specific types of tracing: *temporal traces* and *artefactual traces*. The *temporal traces* show how the traces of time are critically important for displaying attention, whereas the *artefactual traces* show how traces of specific artefacts are equally important to developing collective attention. Lastly, we look at how practitioners are tracing the attention of others when the results of testing do not meet expectations.

Testing practices in a global software development department

In the GlobalSoft setting, testing practices begin in the product descriptions that create the link between requirements and tests. The test team typically varies in size, but is never larger than a total of 12 people across Denmark and the Philippines. The initial work of testing consists of deconstructing the product by analysing the product descriptions and identifying the number of test cases necessary for each product. The Danish test managers perform this task in the early stages of the project before any testing begins. The next step is to develop a test analysis, which is how the testers try to ensure that all of the requirements will be tested. However, this analysis will never be 100 per cent accurate in the early stages of the development because software products are likely to change during the development stage. Therefore, the design analysis is subject to change over the course of the project but is nevertheless necessary as a guideline for the testing. Test managers typically test the design at the Danish location, but on one occasion the Philippine test manager did this task. The test design outlines the purpose of the test and links it to a specific requirement in the product description in order for every aspect of the code to be tested. This is a carefully structured practice of linking every single requirement description to a specific test case to ensure that all is covered. The test design is a document that eventually will be implemented into the test manager tool to capture the documentation of the testing. The document will be placed in particular folders on a shared server with clear links to other important documents, such as the requirement specifications and production descriptions. The test manager tool, the test design, and the

product descriptions are available on the shared server, which developers from both locations have access to. This server is the physical location dedicated to all the data in the project, such as the code or the requirements. The test manager system is used for interacting with the team foundation server, which is a collaborative platform for managing software development projects.

The test manager system allows testers to create and manage test plans; to create, author, and maintain manual and automated test cases; and to file information on bugs associated to specific codes. The test manager system can use data on the server and link to the code, requirements, and tests within an integrated groupware tool shared across the test team. The test manager system is therefore a tool for systematizing and coordinating tasks. For example, each test case can be linked to specific requirements on the team server. In an optimal scenario, every test case will be linked to a specific requirement to create traceability and transparency in the tests. The testers write test cases based on the product descriptions that describe each step required to perform the test. Ideally, a test case only tests one particular aspect of the product, which is also why it is important to link back to that exact aspect in the product description. During our time in the project we observed *unit testing*, *large systems integration testing*, and *black box testing*. Unit testing was performed either by a developer or a tester on a single machine while large *systems integration tests* involved testing the modules together. In this paper, we are going to focus on the *black box testing* that is also described by the practitioners as ‘high-level testing’.

Both the Danish and Philippine testers perform high-level testing in the test manager system. Often testers located in same office had more in-depth knowledge about the software code and, thus, it was important for the testers to coordinate their efforts across geographic sites to ensure quality of the final product. A tester performs a test by following the instructions for each step and verifying the expected result. If the expected result is achieved, the step is passed; if not, the step fails and a bug is reported. The test manager system can assign bugs directly to the developers, and these are usually assigned to the lead developer. After he has checked to make sure the bug has not already been reported, the lead developer then passes on the bug assignment to another developer. Ensuring that each of these steps are correctly described and linked to in the correct product descriptions is a collaborative effort of identifying the proper traces that require close attention to the task.

Two types of tracing attention

Testing is typically accomplished using collaborative systems that manage the process of testing, and coordinate tasks between testers and between testers and developers. However, managing tests is a complex and difficult task in globally distributed teams and traceability is key in coordinating the task. Interestingly, we found that one critical part of traceability was developers' ability to monitor and trace the *attention* of collaborative partners despite being located remotely. In example a test manager located in Denmark expressed that the quality of testing is closely related to the *degree* of traceability in the test case.

This means the ease in tracing the *attention* of his remote colleagues to figure out whether each test case had been correctly linked directly to a requirement in the product descriptions, thus ensuring the quality of the product. Zooming in on how people traced the attention of remote colleagues made us paying heed to different types of traces, critical for the collaboration and directly related to the ability of monitoring remote employees' attention. These were different types of tracings with contrasting characteristics, but they were both related to a diverse set of collaborative technologies and to how these mediated the collaboration. Typically, such technological tools were collaborative source code management systems such as the test manager system, but other tools were also used.

The two different types of tracing are *temporal tracing* and *artefactual tracing* and have distinct characteristics. *Temporal traces* are those that capture time in different ways, allowing for collaborating colleagues to gain insight into the attention of remote workers. Understanding the use of time in terms of *when* collaborators completed an action and where remote colleagues are *currently (at that time)* directing their attention is important for developers to monitor the current state of work and act accordingly. *Artefactual traces* are left behind by remote collaborators through different types of artefacts such as documents, images, videos, etc. – tools used by others to gain insight into where the attention of the remote colleagues was *previously* placed when creating these traces. The information most important to temporal traces is time (past or current), which is used as a vehicle to determine future relevant activities. The information most

important to artefactual traces consists of insights into the content of tasks and the status of these tasks.

In practice, these two types of traces relevant to monitoring colleagues' attention are intertwined and combined. However, the distinction between them is essential for analytical purposes, since both types of traces are important to gaining insight on the attention of collaborators, and we will now look more closely at each one.

Temporal traces

In investigating our empirical material it was clear that the testers in both locations directed their attention towards temporal indicators, such as online presence or time spent on executing a test, in order to trace the direction of attention on a specific task. To provide examples, we observed a situation in which the test manager located in Denmark noticed with whom and when his colleagues were online using the test manager system. Examining our empirical data, we found that the test manager system provided important insights into the temporal traces of remote colleagues attention. During an observation session we engaged in the following conversation with a test manager in the Danish office:

“Interviewer: How do you know that they (the Filipinos) are currently running tests?”

Tester in Denmark: I have talked to [Tester Name] about it earlier today. But I don't know... they may have gone home by now since it is almost 8 p.m. [Philippine time]. No, they are still sitting in the office. [Test manager Name] is still there and so is [Tester Name]. Has [Project leader Name] gone home? He is usually always in the office.”

What we see in the above quotation is that the developer located in Denmark is very aware of the time, as well as the presence or absence of his remote colleagues. He takes these factors into account in deciding how to not interfere with or obstruct their work. His intention at the time had been to demonstrate how to run a test case, but he had to be aware of the work status of his remote colleagues because the test manager system does not facilitate people performing the same test cases simultaneously, and he would risk disrupting his colleagues' current work by doing so. He knew they had attention directed towards and were acting upon a particular part of the system, and thus had to act accordingly.

However how did he know that? By tracking how he traced the attention of his remote colleagues we find that he was aware not only of who was online, but also had knowledge of who is typically online until late. Moreover, he told us that he had corresponded over email with one of the testers online at the time, explaining which tests the testers at the Philippine site would perform that day. Combined with the information provided by the test manager system, he assumed that his colleagues were still running tests. Note that in this case neither the email nor the visual representations on the screen alone would have been enough to conclude that the Philippine testers were actually performing tests. Understanding the existing synchronous status of remote partners was important in determining where and when to do particular activities. Even though remotely distributed testers mostly work asynchronously due to the time zone difference between Denmark and Philippines, they still need to know and be aware of each other's activities. This shows how tracing people's attention can be a practice of

combining knowledge of remote colleagues' work patterns and traces found in emails. But we also observed how other types temporal traces were detected in the test manager system. For example, we observed how developers would refer to temporal traces of each other's attention during interaction by acknowledging each other's work at other times. This was possible since each activity done within the test manager system was annotated with the tester's name, the time, and where the activity had been conducted. During an observation at the Danish location, we observed that a tester located in the Philippine office mentioned that he had noticed a tester from the Danish site had logged into the test manager system late at night (according to the time zone in Denmark). By acknowledging that the work was conducted outside the normal work hours, the developer in the Philippines acknowledged the extra effort and attention to the task by the developer at the Danish site. During the meeting, it was clear that the acknowledgement of presence late at night in a remote local context was seen as demonstrating the commitment to the collaborative engagement, as well as to the task. Thus, leaving traces within the shared IT systems, the testers at the Danish site demonstrated their commitment toward the shared task as well as toward their remote colleagues, without particularly considering or intending this aspect of their actions. This demonstration of commitment without explicit consideration made the collaboration across the remote employees stronger, since all were aware that others made extra efforts to make the shared task successful. However, it was not the temporal trace alone that served in cementing the collaboration. The active effort of acknowledging that the IT developers located in the Philippines

had *seen* and noticed the extra work effort and made this information public – even in a subtle way, by mentioning this incident during a meeting – served to create collective attention.

Tracing the temporal activities of remote colleagues was not only significant in relation to the traces left automatically within the test manager system and brought up in meeting. Also, we saw that temporal traces were important in terms of determining whether remote developers had spent the required time and attention to ensure quality testing. When developers run tests within the test manager system, the times of execution and results are captured. This means that it is possible to get information about when a test was executed, how long it took to run the test, and what were the results. The time it takes to run a test is important information in determining whether the test results are correct or whether there might have been problems with the test's execution. In this way, the temporal traces can give indication as to whether the developer executing the test had carefully enough paid attention to the task. During one observation, it was clear that the temporal traces of execution arrived in unusual patterns for verification. They arrived in bundles, which created uncertainty for the remote colleagues about whether the test was valid or flawed. The testers began to further investigate the situation to determine the attention of the tester and found that the local context had created some unforeseen problems. The office in the Philippines decided to implement scrum-like processes in the development phase and created a backlog based on the products they were expected to deliver, while holding daily sprint meetings. This change turned the three-week iteration cycle into a

two-week iteration cycle that put extra pressure on the testers located in the Philippine office. The main reason for the increased pressure was due to the fact that products that were not tested and fixed until towards the end of the sprint would be transferred to the next two-week iteration. To accommodate these changes, the testers in the Philippines began to have ongoing tests during the sprints to catch as many bugs as possible before the end of the sprint. However, the Danish office was not aware of this sudden change in the coding processes due to some misinformation. Consequently, the changes created unorthodox traces in the test manager system when bundles of logged test cases piled up in the test manager system. In this case, the temporal traces were interpreted as interruptions or miscommunications in the workflow. The sprint testing was different from the official testing procedure and the testers could track uneven patterns that were characterized as logged in large bundles.

When only looking at the digital traces in the test manager system, it became clear to the testers that they did in fact not have a shared directed attention across sites. To fix this issue, the team had to track traces of attention outside of the test manager system to figure out what was going on. In this example, they decided to re-coordinate and re-plan through email and conference meetings in order to take concrete steps to get back on track. The main finding of these meetings was that some testing practices, put in place due to the implementation of new processes at one site, created unorthodox temporal traces of attention that disrupted the coordination across sites, and, thus, extra efforts to restore collective attention were required.

The test manager system also showed the time spent executing a test, and testers would use this information to determine the degree of attention from remote colleagues. In one example, the time captured was as short as 14 seconds, which made the remote testers concerned about the quality of the test because a standard test would normally take much longer to execute. Examining the source of this strange temporal trace revealed that the testers in the Philippine office ran the tests in separate program (a Microsoft Excel sheet) and then had copied and pasted the results into the test manager system. The problem was that such execution of tests was not considered as following formal protocols for proper test performance. The reasoning behind using Microsoft Excel was that the testers in the Philippines were unable to use the test manager system due to bandwidth issues. The bandwidth in the Philippines was only 5 m/bit, causing a lag when trying to update new information in the test manager system. Thus, the testers found the test manager system almost unbearably slow. By applying the use Microsoft Excel to write the test cases and capturing all the data prior to uploading these cases in the test manager system was seen as a feasible solution. It is not our aim to judge whether this practice was problematical or justified, our point here is that the execution time is an important temporal trace in global software development. Monitoring each other's temporal traces was not simply done in terms of one side monitoring the other, but was a mutually shared activity, wherein both parties monitor each other. All testers used the online notification system embedded within the test manager system as an indication of attention towards current tasks. Similarly, a test case performed in only 14

seconds demonstrates a lack of shared attention that needs to be repaired outside of the system itself. The temporal traces in the test manager system give the testers insight into where people have their attention directed, as well as the degree of attention towards a task, thus providing a way to demonstrate people's level of commitment.

Artefactual traces

When examining our empirical data, it became clear that *artefactual traces* were part of identifying the direction of attention among the testers. The artefactual traces were used by the testers as indicators of attention or lack of attention in the collaborative work. The ways in which artefactual traces were used during testing activities indicate the direction of attention among the testers in the global collaboration. One type of artefactual trace that we observed was the use of screen dumps of the tests. The test manager system is capable of making both video recordings and screen dumps directly linked to specific bugs. The screen dumps were used most often to direct attention towards a specific error and would, in most cases, clarify the specific problem for either the developers or the testers who were to later find a solution to the problem.

“Whenever I find an error I can take a screen dump using the feature in the test manager system and it will automatically be linked to the test case. This allows the developer to see exactly what happened during the test. Documenting the specific errors is very important to do correctly. Screen dumps and possibly a comment will ensure the documentation and the developer does not have to go through each and every stage of the test case to track the error.” (Test manager, Denmark)

Screen dumps were useful when testers found a bug and wanted to make sure that the remote developers understood the exact nature of the error. Interestingly, the testers used artefactual traces such as screen dumps or video recordings to determine the direction of attention of their remote colleagues and determine the degree of commitment towards a task. Unpacking how the testers used screen dumps among themselves to trace their attention towards the tasks, we found test managers in Denmark assigned to verify the test cases asked the Philippine testers to make screen dumps of each step in the test case, so the tester in Denmark could follow the testing process in full detail despite being remote.

These screen dumps thus acted as artefactual traces that supported the work in identifying remote colleagues' attention by recording the remote colleagues screens as they interacted with the test manager system. Adding screen dumps provided additional information important to interpreting the results of the tests.

While the test manager system could only display a 'passed' or 'failed' step in the documentation of the execution of the test case, the screen dumps provided extra information of the actual interaction with the system as it happened during the testing. Thus, the screen dumps illustrated the exact result (as in printed on the screen) that occurred during execution of the test, whereas the test manager system only displayed the result in terms of 'passed' or 'failed'.

In the situations when one or more screen dumps are missing the remote tester lacks the necessary information for tracking the test. A missing screen dump in a test might be interpreted as lack of attention towards the task, which might jeopardize the quality of the test. It is unclear whether the test passed without

bugs or if there was something wrong since it lacks the screen dump. To restore the situation additional information is required, and testers will engage in work practices to identify and trace this information. The lack of screen dumps creates an open space for interpretation and the tester cannot be certain of the results.

Thus, the artefactual traces along with the temporal traces represent indications of the direction of attention in a specific situation and when these traces are missing the degree of commitment to the task can be questioned.

Artefactual traces also emerge in the use of documents. It is essential that every requirement in the product description is tested and documented to ensure the quality of the test. Securing quality is partly achieved by creating transparency and traceability in each test case by linking each test case to a specific product description on the shared server. The traceability is enacted by the direct link to a specific requirement in a product description, which then becomes part of the verification and documentation of the test. Transparency is supported through the links by clarifying what is actually being tested in particular test case. One test manager described the importance of links in the following manner:

"[...] explaining to my colleagues the importance of beginning with the test design to outline the test cases, the test design clarifies what is being tested, the purpose of the test, and includes links to the specific requirements that are being tested, so you can trace and make sure that everything has been tested." (Test manager, Denmark)

The links to the product descriptions also reveal something about the degree of detail in the test. If each requirement has a link it is easy to track each test case to a specific requirement. As a general rule, only one particular aspect of the product

can be tested in each test case, which means the links have to also be very specific.

“Let’s say that you have 40 pages of documentation, where each and every bullet point has to be linked correctly to a specific test. Every bullet point represents a requirement which must be implemented in some sort of a test and that is simply not possible in such a short time. Then they [the testers in the Philippines] tell me the next day that they have done. But I can see that they have only done it on the section level, but for every section there is another 10 sub-items. You are not testing 10 different things in a test case, hopefully you are only testing one specific thing.”
(Test manager, Denmark).

It is not enough to just link to the product descriptions to the test case, the link has to be a specific bullet point within the product description. The test design of a product is considered detailed if each and every requirement in the product description is directly linked to a specific test case. However, if the link only refers to the product description as a whole – not to a specific requirement in that document – then the test case is less detailed and specific. The risk is that the test case becomes unclear because it is not apparent what is being tested and what is not tested. To this point, the links between the product descriptions and the test cases make up part of the important artefactual traces used by the testers to determine the attention and thus commitment to detail by their remote colleagues.

In the testing phase, digital traces, such as the product descriptions, are not the only factors presented as artefactual traces in the test manager system. Sometimes testing requires the testers to focus their attention towards the documentation to be able to write the test cases. A large part of testing and writing test cases relies on having knowledge of the system that needs to be tested. You have to know

what is expected of the system to be able to write the test cases. Often the tester has to spend time trying to envision how the system is going to react. Therefore, focusing attention on reading the documentation (i.e., the description of the system) is important to be able to write test cases. In the DAFIC project there were more than 5,000 pages of documentation, making trying to grasp only a part of the information a lengthy and difficult task. Reading does not produce any digital traces that can be observed in a geographically distributed collaboration.

This can create a situation where it is difficult to determine the direction of attention. Internal pressure to ‘prove’ your attention towards particular aspects of the task may arise. For instance, if managers require visible results such as a certain number of finished test cases it pushes the testers to begin writing the test cases, even if they have not read all the needed documentation. In one example, a test manager in the Danish office told us the following:

“They [the testers in the Philippine] are under a lot of pressure – there is always someone measuring how much time they spend on specific tasks. And just saying ‘I am reading documentation’ may lead to the response ‘But you were also reading last week?’ Well sometimes it takes two to three weeks to read the necessary documentation.” (Test manager, Denmark).

Producing written documentation is a much more tangible way to show progress than reading; thus, testers would sometimes prioritize the writing. Much like misinterpretation of silence, the lack of digital traces can blur results and misinform geographically distributed colleagues about where or how remote collaborators are placing their attention. Such situations might be infused with doubt and confusion. This is especially a problem when you are not sitting next to your colleagues and can actually see what they are working on at the moment.

Thus, it requires the practitioners to trace the direction of attention by sending emails or having meetings online or otherwise risk losing track of the process without maintaining continuous attention.

Tracking attention traces

We have presented two types of traces (temporal and artefactual) and shown why tracing the direction of attention is challenging, but nevertheless critical, for testing practices. In everyday work practices, traces provide indications of people's direction towards specific tasks, in combination with practitioners' use of these traces to understand the direction of attention of remote colleagues. The quality of the tests is largely bounded to the traceability and transparency of the test case in the test manager system. In other words, the traces only become important when they are enacted in the everyday work practices of testing. This task, however, can be complicated and difficult, especially when the testers are not located in the same office.

In the following example, we show how temporal and artefactual traces are enacted to track the attention of other testers. These traces are often used in combination, and in practice the testers make no distinction between temporal and artefactual traces. In the example described here, the test manager was performing a test based on a test case written in the Philippine office and was following each step meticulously to verify the test case. At some point during the test, the test manager stops and says:

“In this case, we would prefer to have the values defined for the ‘input’ field. It says ‘input values for the field and click the add button.’ Does this mean that I can write anything [any value] or what? Here the values that need to be added should have been defined. I happen to know that it will work and when you are done you would press the ‘end iteration’ button.” (Tester, Denmark)

The tester had identified a possible problem with the one of the steps in the test case, namely that this particular test case had a step where the tester is asked to type a value into a field. According to the test manager these instructions were imprecise, since the value was not clearly defined. In practice, the user could type any number in the field, even though the expected input was a positive integer in the system. This means that, even if the system works as intended, the user could experience an error since calculations might be wrong. In this case, the information provided in the test manager system was not accurate enough to determine the quality of the test. Ideally, the value should be defined according to the descriptions in the product description and linked directly to the test case.

However, the test case may still be correct if *any* type of value works in the field, even though it was supposed to be constrained inputs. The main problem with the lack of a defined value in this case is not the poorly executed test case, but that the tester could interpret the non-accurate test case as an indication of lack of attention from the tester who had written the case. Without more information about the actual attention directed at the task, it is impossible to identify whether this is sloppy work or could have indeed been done well.

Possibly, the tester who wrote the test case did not understand the importance of a defined value. The tester who wrote the test case was maybe not aware of or forgot to pay attention to the fact that the value of the number could be a problem

when performing the test. He may have thought that this would not be a problem or may not have thought about it at all. On the other hand, the tester may already have prior knowledge of the software to be tested. Often testers have very detailed knowledge of the products that are developed in their own office, since they are located near the developers and often participate in the same meetings. In that case, the tester may have had knowledge about this specific field and the range of values; thus, the tester was aware that this did not require a defined value to work. Instead, the tester may have known that any type of number was valid in this particular field and, thus, there was no need to clarify in more detail. In this example, the test case may be perfectly valid but since the digital traces leave room for misinterpretation the test manager can become unsure about the conclusion. In other words, the test manager and the tester do not have their attention directed towards the same details. It may not be ‘wrong’ *per se*, but the digital traces in the system only tell part of the story. The test manager cannot predict the direction of attention based solely on the digital traces in the test system (i.e., the lack of definition in the value section). Instead, the test manager needs to trace the direction of attention by identifying other temporal and artefactual traces. As an example of how this tracking took place a test manager said the following:

“It is typically something that I can sense without being able to pin point exactly what it is. But for instance it occurred while checking some test results in the test manager system. Beforehand we had agreed that they [the testers in the Philippines] would supply each step in the test with a screen dump, so I was able to track what had happened in the tests. But if I then see that only every third step has a screen dump attached, it becomes difficult to determine how the test has

been done. Or you can see that some of the more difficult steps in the test case – those that can cause doubt – they may not have done them at all.” (Test manager, Denmark).

The example demonstrates the importance of mutual attention for interpreting the test cases and ensuring the quality. We found that while the practitioners used the traces in the test manager system in their work to determine the direction of the attention among their colleagues this is only part of the work. Artefactual traces such as screen dumps or video recordings were commonly used in combination with the temporal traces in the test manager system to aid the process.

Initially, the tracing work was *ad hoc* and individually based; consequently, the testers could not sufficiently trace the direction of attention between their colleagues located in the remote office. During the later stages in the project, constantly tracking people’s attention across sites became increasingly important, and one of the changes was the introduction of daily agenda-less video meetings with developers from both locations. These meetings would be held at 10:00 AM (Central European Time), which was late afternoon in the Philippines and early morning in Denmark. The number of employees partaking in the meetings would vary slightly, but typically the meeting would include one to three persons from the Danish test team and a similar number from the Philippine test team. The meetings lasted between five and 30 minutes, depending on existing issues. From the perspective of tracking people’s attention, the meetings supported the testers in keeping up with tracing each other’s progress by informally updating each other on individual work. In other words, they used the meetings to make visible their attention, making it possible for others to trace the direction of attention

among their colleagues. The agenda-less meeting created a space of interaction where the project members could acknowledge each other's work by directing their attention toward the work effort of remote partners and also could address specific critical work practices in the test manager system. For instance, openly acknowledging people who are logged in late at night to execute some tasks in the test manager system, which also provides information about the remote testers dedication to finishing the task. The daily meetings did not only facilitate a shared attention during the meeting they also created a foundation for maintaining that attention throughout the day.

"[The Philippine developers] write a status email when the workday ends where they make an assessment of the accomplishments of the day. For instance, that they expect to have finished 80 per cent of the tasks and write a list of the upcoming tasks. Or, for example, that they expect to create test data in the next week or that they will perform preliminary tests. Or whatever task that seems relevant at the time." (Test manager, Denmark)

The testers would return with more queries or answers to questions raised in the meeting the same day through other means of technology use. As one test manager told us during an interview: *"Later during the day, we typically have some interaction using Lync [a messenger system] where we are writing together and working towards a common understanding."* As we can see the testers would track each other's attention throughout the day using daily status emails or questions regarding specific tasks, which served as indications of their current attention and where it was directed. Table 2 summarizes the findings.

Table 2: Summary of empirical findings.

Traces	Description	Examples
Temporal traces	Captures time in different ways relevant for the collaborating colleagues. Displays time, in terms of <i>when</i> collaborators did an action, as well as where remote colleagues <i>currently</i> is directing their attention	Online presence Bundles of test cases Quick testing
Artefactual traces	Different types of artefacts such as documents, images, videos which are used by developers to get insights into the status of the tasks currently undertaken by others	Screen dumps Links to product descriptions Video recordings
Tracking attention	Traces only become important when they are enacted in the everyday work practices of testing. Employees are tracking attention using the range of different tools and systems, thus creating a multiplicity traces to determine the direction of attention of their colleagues	

5 Demonstrating commitment

In our effort to understand the ways in which IT developers demonstrate commitment in a technology-mediated environment, we found that temporal and artefactual traces were critical for understanding the strategies individuals used to demonstrate commitment as part of the collaborative work. Previous research (Bjorn & Hertzum 2006; Simone & Divitini 1999) refers to commitment as the discursive action of declaring commitment to a task in combination with the degree of commitment. However, we found that, to fully comprehend commitment as a feature of collaborative work rather than as a declarative state of mind, we needed to move from the perspective of *declaring commitment* towards the perspective of *demonstrating commitment*. This rephrasing transforms the basic nature of what commitment is and how we could investigate it as part of

collaborative practice. In our work, it is clear that commitment is not only a mental or discursive construct (Allen & Meyer 1990b), but also part of the everyday practice.

Identifying strategies for demonstrating commitment especially in geographical distributed work situations are thus important, because these affect how collaborators manage their relationships. Previously, bodily presence has been identified as the highest degree of demonstrating commitment in a geographically distributed setting, because it demonstrates a willingness to spend time and effort to travel at other locations (Nardi 2005), however this strategy does not always apply. Instead we found that strategies to demonstrate commitment through technology-mediated means within a distributed work setting were associated with two types of traces captured by the diverse set collaborative systems used by the employees – temporal and artefactual traces. The traces revealed a pattern of behavior that could be used as a deduction of where the attention of remote colleague was or had been directed during activities related to solving tasks.

Seeing your colleagues online at odd hours in the test manager system was a testament of their current or previous attention to the work, which was used to assess the commitment of others. However, not only the visibility left in traces formed the strategies of commitment. Rather, we found that the *mutual acknowledgement of the existence of these traces* was also essential. Thus, it was the reinforced and maintained acknowledgement of noticing traces, such as during collaborative activities like daily video-mediated test meetings, which

linked the traces to the demonstration of commitment. We saw how the testers combined information on online presence in the application with information received in emails to conclude that their colleagues were still running tests at a given time. Similarly, we observed how test results in the test manager system were coupled with screen dumps as well as screen recordings – two types of traces that together indicated the direction of attention towards a task.

One important result from our analysis was that the practice of demonstrating commitment could not be understood outside the context of the task. This means that demonstrating commitment was always directly related to certain tasks, and tracking the traces of people's commitment would be different dependent on the task. Determining the commitment of remote testers by their dedication to details when executing tests was different than determining the commitment of remote testers in terms of tests following unorthodox patterns. In this way, demonstrating commitment was not simply about being available for remote workers at certain times. Demonstrating availability is important for commitment practices, but we found that bodily presence was not the predominant strategy to demonstrate commitment (Nardi 2005). Instead, we found that the digital traces revealed not only other people's presence and availability, but also how they performed certain tasks.

The use of coordinative artifacts as an approach to demonstrate commitment has been pointed to by for example research within agile development or lean methodologies (Whittaker & Schwarz 1999). Here, shared physical artefacts such as scrum boards (Cohn et al. 2009) or war room posters (Bjørn 2011) play an

important role supporting software developers or engineers in declaring in a discursive manner the commitment to certain tasks and deadlines, while making these declarations tangible through Post-it notes and practices of constantly recording activities at work (Esbensen & Bjørn 2014). Many of these studies point to the importance of physical proximity and suggest strategies of co-location as critical for making the collaboration function well (Mark 2002b). Investigating the strategies by which the testers in the DAFIC project demonstrate commitment in the globally distributed setting, we found that the collaborative systems served as a way to which the testers could capture their own while monitor other's traces of work. In this way, the diverse ways in which the technologies were used – through the use of screen dumps, video recordings, video-meeting equipment, test manager systems, email, etc. – all acted as one connected collaborative artefact in terms of enabling testers to demonstrate commitment. However, the connections and associations across all these different applications, which were critical in establishing collective attention, could be seen as a multiplicity of collaborative artefacts supporting commitment practices at work. This work of tracking associations and connections based upon the temporal and artefactual traces in several collaborative technologies required extra work by the developers. Nevertheless, it was essential to the ways in which the collaborative developers established collective attention, thus creating the opportunity for demonstrating commitment in a technology-mediated globally distributed collaborative setting.

6 Conclusion

In this paper, we set out to unpack commitment as a practice of collaborative work and develop the basic foundations for how commitment can be understood in the work place. In particular, we were interested in understanding how commitment was enacted within collaborative work in which employees are geographically distributed, yet closely coupled in their work. Closely-coupled work structures supports globally distributed work (Jensen 2014; Bjørn et al. 2014), nevertheless nobody has previously studied how commitment can be demonstrated within such work settings.. We found that existing conceptualizations of commitment were not able to explain how commitment was accomplished, and thus we had to re-think fundamentally how commitment might emerge within global work. We wanted to take a different approach and investigate what it would mean to think about commitment *as a practice*, rather than as a discursive construct. By applying the practice approach to our investigation of commitment, we argue that commitment is not a state of mind, but instead a practice interlinked within collaborative activities, and as such is part of the invisible work that makes collaboration function. Analyzing empirical observations from our ethnographic data, we found that commitment practices centered on tracing remote colleagues attention through temporal and artefactual clues captured by technology mediation. This tracking of traces where attempts to establish collective attention concerning the task, the process, and the people involved, thus providing important information about remote colleagues. We

found that the work practices of tracking traces were not only about knowing the availabilities of others, but also concerned the ways in which individuals were able to demonstrate commitment and how these practices of commitment at work became part of the collaborative work arrangement.

7 Acknowledgements

References

- Allen, N.J. & Meyer, J.P., 1990a. Organizational Socialization Tactics: A Longitudinal Analysis of Links to Newcomers' Commitment and Role Orientation. *Academy of Management Journal*, 33(4), pp.847–858.
- Allen, N.J. & Meyer, J.P., 1990b. The Measurement and Antecedents of Affective, Continuance and Normative Commitment to the Organization. *Journal of Occupational Psychology*, 63(1), pp.1–18.
- Bjorn, P. & Hertzum, M., 2006. Project-Based Collaborative Learning: Negotiating Leadership and Commitment in Virtual Teams. In D. V. Greunen, ed. 5th Conference on Human-Computer Interaction in South Africa (CHI-SA).
- Bjørn, P., 2011. Co-Constructing Globally Collaborative Spaces and Places: an Ethnographic Study of How Globally Distributed Engineers Handle Planning Practice. In Scandinavian Conference on Information Systems (SCIS). Turku, Finland. Springer, pp. 16–18.
- Bjørn, P. & Østerlund, C., 2014. *Sociomaterial-Design: Bounding technologies in practice*, Springer.
- Bjørn, P. et al., 2014. Does Distance Still Matter? Revisiting the CSCW Fundamentals on Distributed Collaboration. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(5), pp.1–27.
- Blomberg, J. & Karasti, H., 2013. Reflections on 25 Years of Ethnography in CSCW. *Computer Supported Cooperative Work (CSCW)*, 22(4-6), pp.373–423.
- Cohn, M.L., Sim, S.E. & Lee, C.P., 2009. What Counts as Software Process? Negotiating the Boundary of Software Work Through Artifacts and Conversation. *Computer Supported Cooperative Work (CSCW)*, 18(5-6), pp.401–443.
- Cushen, J. & Thompson, P., 2012. Doing the Right Thing? HRM and the Angry Knowledge Worker. *New Technology, Work and Employment*, 27(2), pp.79–92.

- Dabbish, L., Kraut, R. & Patton, J., 2012. Communication and Commitment in an Online Game Team. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Austin, Texas, USA: ACM, pp. 879–888.
- Esbensen, M. & Bjørn, P., 2014. Routine and Standardization in Global Software Development. In International ACM Conference on Supporting Group Work (GROUP). Sanibel Island.
- Forsythe, D.E., 1999. “It’s Just a Matter of Common Sense”: Ethnography as Invisible Work. *Journal of Computer Supported Cooperative Work (CSCW)*, 8(1), pp.127–145.
- Harper, R.H.R., 2000. The Organisation in Ethnography. *Computer Supported Cooperative Work (CSCW)*, 9(2), pp.239–264.
- Howcroft, D. & Taylor, P., 2014. ‘Plus Ca Change, Plus La Meme Chose?’—Researching and Theorising the “New” New Technologies. *New Technology, Work and Employment*, 29(1), pp.1–8.
- Jensen, R.E., 2014. Why Closely Coupled Work Matters in Global Software Development. In International ACM Conference on Supporting Group Work (GROUP).
- Luff, P., Hindsmarsh, J. & Heath, C., 2000. *Workplace Studies: Recovering work practice and informing system design*, Cambridge: University Press.
- Mark, G., 2002a. Conventions and Commitments in Distributed CSCW Groups. *Journal of Computer Supported Cooperative Work (CSCW)*, 11(3), pp.349–387.
- Mark, G., 2002b. Extreme Collaboration. *Communications of the ACM*, 45(6), pp.89–93.
- Martin, D. et al., 2006. Achieving Dependability in the Configuration, Integration and Testing of Healthcare Technologies. *Computer Supported Cooperative Work (CSCW)*, 15(5-6), pp.467–499.
- Nardi, B., 2005. Beyond Bandwidth: Dimensions of Connection in Interpersonal Communication. *Journal of Computer Supported Cooperative Work (CSCW)*, 14, pp.91–130.
- Randall, D., Harper, R. & Rouncefield, M., 2007. *Fieldwork for Design*, Springer.
- Robinson, M., 1991. Double-Level Languages and Co-Operative Working. *Ai & Society*, 5(1), pp.34–60.
- Rooksby, J., Rouncefield, M. & Sommerville, I., 2009. Testing in the Wild: The Social and Organisational Dimensions of Real World Practice. *Computer Supported Cooperative Work (CSCW)*, 18(5-6), pp.559–580.
- Rusman, E. et al., 2013. The Mind's Eye on Personal Profiles: A Cognitive Perspective on Profile Elements that Inform Initial Trustworthiness Assessments and Social Awareness in Virtual Project Teams. *Computer Supported Cooperative Work*, 22(2-3).
- Simone, C. & Divitini, M., 1999. Integrating Contexts to Support Coordination: The CHAOS Project. *Computer Supported Cooperative Work (CSCW)*, 8(3), pp.239–283.
- Søderberg, A.-M., Krishna, S. & Bjørn, P., 2013. Commitment, Trust and Cultural Sensitivity in Strategic Partnerships. *Journal of International Management*, 19(4), pp.347–361.

Whittaker, S. & Schwarz, H., 1999. Meetings of the Board: the Impact of Scheduling Medium on Long Term Group Coordination in Software Development. *Computer Supported Cooperative Work (CSCW)*.

Winograd, T. & Flores, F., 1986. *Understanding Computers and Cognition*, Intellect Books.

The next paper to be presented is paper 4

Jensen, R. E. (2014). Why Closely Coupled Work Matters in Global Software Development. *Presented at the International ACM Conference on Supporting Group Work (GROUP)*, November 2014, Sanibel Island, Florida, USA.

Why Closely Coupled Work Matters in Global Software Development

Rasmus Eskild Jensen
IT-University of Copenhagen
Rued Langgaards vej 7
2300 Copenhagen
raej@itu.dk

ABSTRACT

We report on an ethnographic study of an offshore global software development project between Danish and Philippine developers in a Danish company called GlobalSoft. We investigate why the IT-developers chose to engage in more closely coupled work as the project progressed and argue that closely coupled work supported the collaboration in a very challenging project. Three key findings are presented: 1) Closely coupled work practices established connections across the collaboration ensuring knowledge exchange and improving coordination between project members, 2) Closely coupled work practices diminished the formation of sub-groups locally and established new faultlines across the geographical distance, and 3) Closely coupled work enabled the creation of connections across organizational hierarchies allowing information to travel seamlessly between layers in the organization and consequently the project members could better anticipate issues and act accordingly. The implications of these findings include a reconsideration of the significance of closely coupled work in distributed settings. Also our findings open up discussions of why closely coupled work matters in global software development.

Categories and Subject Descriptors

H.1.2 [User/machine information]: Human factors.

H.5.3 [Group and Organizational Interfaces]: Computer-supported cooperative work.

General Terms

Human Factors.

Keywords

Closely coupled work; Global Software Development (GSD); Ethnographic study; Computer-Supported Cooperative Work (CSCW).

1. INTRODUCTION

The CSCW community has in the past years added more importance to the collaborative practices of global software development (Jensen and Bjørn 2012, Avram et al. 2009, Boden and Avram 2009). Working in a geographically distributed setting across cultural differences and time zones influence the work practices of IT-developers (Noll et al. 2010, Matthiessen et al.

2014). Overcoming challenges such as coordination (Christensen and Bjørn 2014, Herbsleb 2007) establishing trust (Boden et al. 2009), and managing culture (Krishna et al. 2004) are critical for the IT-developers to successfully manage global software development projects. Researchers have proposed different strategies to overcome these challenges and some have even proposed that complex and highly interdependent software development tasks are unmanageable in geographically distributed teams (Olson and Olson 2000). Instead IT-developers should focus on reducing connections across the geographical distance and minimizing interaction between the developers (Hertzum and Pries-Heje 2011). Conversely this paper presents a case where the IT-developers were able to create a complex software solution while working in closely coupled collaboration and consequently experiencing the best offshore collaboration to date. We found that key to this outcome was the IT-developers ability to shift their work practices from a loosely coupled to a more closely coupled configuration

In this paper we ask: *Why did the IT-developers choose to engage in more closely coupled work?* Answering this question, we provide empirical data from a software development setup between a large Danish company and a Philippine offshore office. In this case the Philippine office had initially been functioning as an outsource software supplier for three years before it was bought by the Danish company. This paper report from the point in time when the Philippine office became an offshore office and events that followed in the next three years. Financially the company had benefitted from having a outsource supplier in terms of competitive power that thus led to buying the Philippine office. However, the collaboration between the Danish and Filipino IT-developers had not been optimal during the first three years of collaboration. There was a lack of trust between the two groups, little to no social interaction and a general feeling of being them and us. Work practices were loosely coupled, tasks were divided among the IT-developers and the collaboration consisted of relatively strict barriers in the collaborative work in terms of limited travel funding and using the Philippine office as a resource rather than a strategic partner (Søderberg et al. 2013).

We observed over the period of three years how the IT-developers slowly transformed their work practices to overcome the challenges experienced in the collaboration. The transition was not carefully orchestrated by the management group but rather carried out as a non-sequential process that involved a range of different employees in the company. Over time the IT-developers facilitated more closely coupled work, which was essential to solve the task. This paper point to three key factors of how the collaboration improved during the transformation from loosely to closely coupled work. Firstly, closely coupled work practices established many connections across the collaboration ensuring

knowledge exchange and improving coordination. Secondly, closely coupled work practices diminished the formation of sub-groups locally and established new faultlines across the geographical distance by intensifying travels, collocating people and increasing the frequency of meetings between distributed members. Lastly, closely coupled work created connections across organizational hierarchies and enabled people to take responsibility and share mutual accountability of the project outcome. Connections across organizational hierarchies allowed information to travel seamlessly between layers in the organization and consequently the practitioners could better anticipate issues and act accordingly.

The paper begins with a description of the research method followed by a section on related work. Subsequently the case is described and the empirical results are presented in order to answer the research question. Finally the findings are discussed and the conclusion is presented.

2. METHOD

The company we investigated is called GlobalSoft (a pseudonym). It was incorporated in 1994 and has its origin in Denmark. GlobalSoft employs more than 1700 people in various destinations such as Denmark, China, Switzerland and the Philippines. The Philippine office was originally an independent supplier of programming resources for GlobalSoft. In 2009, the Philippine office was bought by GlobalSoft, which had 85 employees at the time. The Philippine office merged with an existing department in Denmark.

Data collection was undertaken by three researchers and took place from December 2010 to October 2013. The subjects under study were highly professional and worked with expertise within their fields. We conducted on-site observations in Denmark and the Philippines, shadowing employees, participating in video meetings, and observing everyday practices. The data include 28 audio-recorded and transcribed interviews (19 in Denmark, 9 in the Philippines), each about an hour long. We interviewed, observed, and interacted with developers, testers, IT-architects, project leaders, and managers. We spent a total of almost two months in the Philippines during four different visits (December 2010, July 2011, November 2011, and January 2012), and 12 months in Denmark. During data collection, we discussed initial findings with the workers in casual conversation and in official presentations and workshops. Two researchers held four video-recorded workshops were held in the Philippines and Denmark. The data material such as the presentations and workshops was used to interrogate and validate our findings (Eisenhardt 1989). Lastly, we analyzed the video recording and internal company documents. Our methodological approach was an open-ended study design inspired by grounded theory (Strauss and Glaser 1967). The variety of data collection methods helped establish a grounded understanding of the complexities of geographical distributed development practices. The analysis of the data material required several iterations of re-reading, coding, and write-ups to categorize and connect the themes that emerged

3. CLOSELY COUPLED WORK IN DISTRIBUTED TEAMS

To understand why the developers engaged in more closely coupled work practices one has to investigate the notion of a collaborative task. The CSCW research on coupling of work focuses on collaborative work and the nature of the task. Collaborative work is defined by multiple people being mutually engaged within a common field of work, and where individual

activities have a direct impact on the collaborative partners (Schmidt and Bannon 1992). Coupling of work has been the subject of CSCW for many years (Neale et al. 2004, Pinelle and Gutwin 2003, Herbsleb et al. 2000). Coupling of work refers to the nature and degree of communication needed to solve a specific task (Olson and Olson 2000). A cooperative arrangement involves a range of more or less interdependent activities that can be described as a set of activities (Schmidt 2011). Some of these activities can be characterized as 'loosely coupled' while others can be characterized as 'tightly coupled'. Teams working individually on distinct contributions can be characterized as loosely coupled whereas teams with sequential or reciprocal interdependence in the cooperative arrangement are tightly coupled (Balakrishnan et al. 2011). According to Olson and Olson (2000) loosely coupled work is composed of simple standardized tasks with few interlinked dependencies whereas tightly coupled work, however, is ambiguous and highly interdependent nature and thus very difficult to divide into smaller segments. Software products with many ambiguities and a high degree of dependency among the software components can be characterized as tightly coupled work. Some research propose that loosely coupled tasks are more suitable for distributed work (Hertzum and Pries-Heje 2011, Olson and Olson 2000) which can easily be divided into smaller sub-tasks. Since loosely coupled tasks have few dependencies they can be reassembled again and requires little coordination and interaction across geographical distance (Mockus and Weiss 2001, Grinter 2003). Tightly coupled work should preferably be used in a collocated setting because collocated people typically share a common ground and are better able to solve the ambiguities of tightly coupled work (Olson and Olson 2000). Contrary to this other researchers have suggested that tightly coupled tasks are actually better suited for distributed work because it requires close connections and thus enables the practitioners to engage with each other on a mutually shared task (Bjorn and Ngwenyama 2009). Thus this line of research points to the necessity of closely coupled work as an incentive for increasing interaction and forming partnerships among distributed members (Bhat et al. 2006). Collocated software development requires a lot of coordination (Grinter 2003) and the added geographical distance only increases the need to coordinate (Boden and Avram 2009, Avram et al. 2009). However, closely coupled work in a geographically distributed setting is difficult to achieve and will require changes in the work practices both locally and remotely (Matthiessen et al. 2014). In global software development projects the IT-developers are often dependent on each other in order to solve the task especially in those situations where the final product is applied into a specific domain area for instance when one group of IT-developers has the necessary domain knowledge, which needs to be communicated across the geographical distance (Jensen and Bjorn 2012). Strategies to minimize the need for coordination are segregation and standardization where tasks are divided into smaller pieces with specific requirements (Gerson 2008, Mockus and Weiss 2001). However, standardization and segregation do not eliminate the need for coordination in software development projects because all the separated parts have to be assembled to a functioning software product. The reassembling of the software requires coordination to ensure that dependencies between the small pieces are correct when the all the pieces are collected (Gerson 2008, Grinter 2003).

The increased need for coordination is further complicated in projects that span cultural differences (Krishna et al. 2004). Culture defined as a set of values, beliefs or norms (Soderberg and

Holden 2002) allows for a deeper understanding of how cultural distance impacts distributed teams unlike a cultural understanding based on national categories (Hofstede and Hofstede 2004). Moving away from national stereotypes recent research have proposed that culture can instead be understood as “*a shared web of meanings that shapes roles and interpretations, and is dynamically (re)negotiated by the actors in the course of their daily work*” (Boden et al. 2012). A lack of a shared web of meanings can lead to misunderstandings and collaboration can be further obscured due to emergent faultlines within groups (Cramton and Hinds 2005). Faultlines increase the likelihood of alignment among group member’s demographic attributes, which is the foundation for sub-group formation. Demographic faultlines such as gender, age, cultural background or professional expertise have increased risk of forming in distributed teams. Research has shown that sub-group dynamics can impact negatively on the collaborative work and establish a sense of “us and them” based on ethnocentrism (Cramton and Hinds 2005). Sub-groups within a project can be damaging for the collaboration and result in conflict as other studies of global software development projects have shown (see for example Marrewijk 2010, or Metiu 2006). These studies found that strong internal groupings hindered the collaborative work or even caused the collaboration to end permanently. Negative sub-group dynamics can be mitigated by focusing on creating strong connections across distributed groups that establishes cross-cutting faultlines across the groups located in the different locations. The positive aspects of sub-groups are specifically achieved when faultlines are established between distributed team members that potentially can create cross-cultural learning (Cramton and Hinds 2005).

While the increased need of coordination required for closely coupled work might be challenging in a distributed setting, it may also offset positive aspects in the collaborative work. Research has shown that social translucence, which is key for collaborative work, is a process of negotiation among project participants. Bjørn and Ngwenyama states that: “*Creating translucence and building shared meaning at the work practice level is a negotiation process between the participants where sub-languages are in contact, new languages are formed and meanings are developed.*” (2007 p. 20). Thus participants distributed teams need to engage in on-going negotiation to develop new meaning and establish social translucence. Social translucence consists of three principles for social interaction i.e. visibility, availability and accountability (Erickson and Kellogg 2000). Within the field of CSCW studies of a similar concept, namely awareness has shown how collocated employees display and monitor activities in order to act accordingly (Schmidt 2002, Heath and Luff 1992). Social translucence contains elements that are key to coordinating activities between people. Firstly it gives visibility of our surroundings and enables people to act accordingly. Secondly it establishes a mutual visibility where others can see you and you know that they can see you. Finally, shared visibility creates accountability because we adhere to social norms of behavior and act within the rationale of these norms. In a collocated setting social interaction is seamless and achieved without much effort whereas in a distributed setting the social pressure to reinforce norms of behavior is usually weak (Mark 2002). Thus it can be very challenging to create the necessary conditions for social translucence in distributed teams.

In summary working on tightly coupled tasks in a distributed setting requires more effort due to the geographical distance and cultural differences. To overcome the burden of coordination software development teams often choose to use segregation or

standardization with few connections across the different locations. So why does closely coupled work matter in global software development projects? Interestingly we observed in our case how the IT-developers chose to engage in more closely coupled work as the project progressed. This would appear to be a counter-intuitive solution seemed to help the IT-developers overcome the challenges they were facing in the collaborative work. However, we saw that the trade-off of loosely coupled work is that no shared meaning is developed, the risk of communication breakdowns increase and sub-group dynamics emerge along demographic faultlines. In the following section we are going to present the findings and show in detail why the closely coupled work is important for global software development projects.

4. FINDINGS

This section consists of four sub-sections that identify the work practices at different stages. The sections illustrate the transformation of going from loosely coupled work to closely coupled work practices in a global software development department. The first section describes the situation prior to the project where the initial steps towards offshoring is taken. The second section describes the status of the collaboration in the initial stages and focuses on examples that illustrate the work practices during this period. The third section describes how work practices changed over time as the collaboration transformed to more closely coupled work. The fourth and final section describes the emerging work practices in an offshore collaboration.

4.1 Going from outsourcing to offshore

The focus of this paper is a single department in Globalsoft. The department is the largest of its kind in Denmark and in charge of developing new software systems to primarily Danish clients. The department was able to handle all stages of the development e.g. specification, development, testing and implementation. In 2006 a Philippine outsource supplier was hired to support the Danish department. The main contribution from the Philippine supplier was recognized to be scalability and low costs, but had the potential to contribute with specific developmental expertise. The Philippine supplier was bought by Globalsoft in 2009 and became an official part of the company. The Philippine office merged with the Danish department, so that it consisted of two offices – one in Denmark and one in the Philippines. At the time the Philippine office consisted of 85 employees.

The transition from working in a collocated environment to geographical distributed setting was challenging for the Danish employees. The Philippine office had been working in distributed setting for a longer time and could be characterized as “born global”. The employees in the Danish office were used to do everything by themselves and were thus quite skeptical towards this change, which many described as a nuisance enforced upon them by the management group in GlobalSoft. The collaboration with the Philippine office lacked success stories and many remained skeptical. As one Danish developer said “*There has not been a success story yet. If they existed they would have been talked about. I am certain of that.*” The managers defended the change to offshore by referring to the market dynamics, stating that the GlobalSoft required an offshore component to remain competitive even though the Danish employees had experienced few clearly successful projects with the Philippine office.

4.1.1 Persistent loosely coupled work practices

Initially the work practices between the Danish and Philippine office were still affected by the setup used when the Philippine office was merely a supplier of services and not fully integrated

into the department. The Philippine office was assigned hours according to a fixed-price model meaning that they would estimate the number of hours for a task and give a fixed price for doing the task. This practice carried with it some negative implications for the employees in both Denmark and the Philippines, but to really appreciate this we need to understand the organizational setup between the Philippine office, the Danish office and the clients. The Danish office was responsible for client relations and would typically try to establish a close connection to the clients. In our case we observed that the client came every week to GlobalSoft to spend the day with the project members in the office. The employees in the Philippine office had no contact with the client at any time during the project. It was a deliberate strategy from the company to keep the offshore component hidden from the client and leave the impression that the product was the same regardless of the organizational setup.

The organizational setup with the fixed prices created on-going discussions about change requests and whether they are legitimate changes or an error caused by the department. If the change request could be characterized as legitimate change then the client would have to pay for the extra hours needed to fix the problem whereas errors put cost on the project budget. Since the Danish employees had the client contact they also had to negotiate change requests with client. But with the fixed price setup the Danes were now in a position where they had to negotiate on two front namely with the client and the Philippine office. In the days before outsourcing to the Philippines this issue had been less of a problem, because the Danish office only had to negotiate with client to solve a problem. The fixed price setup consequently required negotiation with both client and the Philippine office when neither of them was willing to pay the extra cost of a change. Moreover some employees in the Danish department would state outright that they did not trust the estimates coming from the Philippine office and several of the Danish employees we interviewed suspected Filipino employees of padding time estimates. Danish employees felt that within this arrangement the Filipino workers did not share full responsibility for outcomes of the projects, and were only concerned with covering their estimated hours. Moreover at the time the employees in the Danish and Philippine office shared little insight into each other's work conditions because intra-organizational travels were limited by relatively strict company rules. Organizational constraints at the time limited the funding for traveling between Denmark and the Philippines since travels were not part of project budgets. Funding for travels had to be granted by the vice president in the department and the requests would often be denied. Travel requests were denied because the gain of face-to-face interaction was difficult to determine and weighed against the cost for the project often led to rejection.

4.1.2 Working as "us and them"

The work practices in the collaborative work seemed experimental and haphazardly implemented according to the employees. As a Danish IT-architect told us during an interview:

[...] The projects I have been involved in were more or less like shooting buck shots at the problems when these urgent needs occurred. Like putting out fires – you just do something, right? But we have not yet had the discussion of how the basic premises of this collaboration are going to work.

The past three years of experience with the Philippine office had been dominated by a trial and error approach. The problem was that mistakes were repeated in new projects. One example of this was a model for global collaboration that was introduced in the

department. The employees in the Philippine office received training in this new model that specifically described the roles and responsibilities for global collaboration. However, the Danish employees were more reluctant to accept this new model and kept using the work practices they had used in previous projects. The implication of one group of employees working with one model in mind and the other group working differently caused some frustration for the Philippine employees since they had spent time and effort in order to learn the new global work practices. The Filipino employees were generally well prepared for working in a global context in terms of language skills and their use of CMC-tools such as instant messenger. On several occasions we observed them actively engaged in conversations on instant messenger even with people located right next to them. They were however concerned about the lack of awareness of the "global component" as one Filipino manager said:

Because a lot of it isn't culture specific. It's really problems with time zone, problems with, well not problems, but rather just people having to get used to be more professional and people having to realize that they have a global, there's a global component in the work that we're doing. Not just on the Danish side but also on our side.

Clearly, both the Danish and Filipino developers felt that the collaborative aspects of the work could be improved, but they identified different challenges as the root of the problems. Filipino employees generally felt that the Danes lacked sufficient English skills to master geographically distributed work, which was evident when Danish employees were reluctant to call directly to their Filipino colleagues. Moreover the Filipinos expressed a concern of being excluded in the collaboration and felt a "lack of interest" from the Danish employees. One example of this, which happened on several occasions, was emails written in Danish that was sent to the Filipino employees. For instance a string of forwarded emails where the first email would have a sentence like "Important! See below" and then the rest of the emails would all be written in Danish. The Danes, on the other hand, had little trust in the technical expertise of their Filipino colleagues and often perceived themselves superior in terms of experience. Arguably the Danes largely had more experience because they were typically older and not hired straight from the university. In the Philippines promotions was sometimes used as an incentive structure for the employees to make them stay at the company. Of course this would also be the case in the Danish office to some extent, but the competition over skilled employees was much more present in the booming offshore market in the Philippines. As a result the Philippine developers got promoted faster than their Danish colleagues and typically had less experience. For instance, one who had approximately two years of experience held the role as developer lead in the Philippine office which is very little compared to the Danish developers average experience. Thus the people who held the project roles in the Philippine office often had a lot less practical experience compared to the Danish employees. Consequently the employees in the Danish office sometimes expected more from the Philippine employees based on their position and not their actual experience. The experience gap was sometimes ignored or forgotten in the project planning and thus expectations to the technical expertise of the Filipino developers failed on several occasions.

4.1.3 Still working as an outsourcing company

The situation caused some serious constraints on the relationship between the Danish and Filipino employees, which heightened by the relatively strict travelling rules at the time. Less travelling

meant that employees in the projects rarely met and had little change of building strong social ties with each other. The challenge that was most prominent at this stage was the general feeling of “us and them” in the projects. This feeling was shared in both offices and exemplified in the little social interaction that happened at this stage. Locally people had casual interaction, shared dinner together and attended social activities but globally no shared activities occurred. The work practices were still primarily based on an outsourcing setup with clear distinctions of them and us, both in terms of social relations but also the division of work and the economic setup between the two offices.

4.2 The initial stages of the project

The department won a call for tender for a big project during this period of transforming from outsource to offshore. This project was thus the first major project with the Philippine office as an offshore component and the changes in the project illustrates the general transformation of practices in the department. The project was the largest development project undertaken in the history of the Danish-Philippine department. The contract for this particular project was signed late 2010 and the preliminary preparations took place in spring 2011. In late 2010 the department undertook its biggest task to date when it won the call for tender of a large project for the Danish government. It was going to span more than three years with the largest budget to date and require technical competences that the department did not have at the time. However, no one in the department or the client was fully aware of the scale and complexity of the project at this moment in time.

The fixed price model had been changed meaning that the both offices shared the responsibility in the project. The Philippine office was now integrated into the project budget along similar offices in the company. The discussion of change requests ended because everybody was equally responsible for fixing errors. Travels no longer needed managerial approval and each project had a travel budget allowing the project leader to make flexible decisions regarding travels. This made travels between Denmark and the Philippines possible to a greater scale. The changes that came from the management group were based partly on the feedback received from the employees in the department.

4.2.1 Changing the scope of the project

The work practices at this stage had a great impact on the outcome of the project primarily because the scope of the project was changed. The project contract signed stated that the Philippine office was fully responsible for the development of the product to the Danish municipalities. However, the people who had authored the contract left the company and an experienced IT-architect from the Danish office was assigned to write the requirement specification to the Philippine office. During this task the IT-architect from the Danish office apprehended the complexity of the task both in terms of technical expertise but also the amount of interaction with public institutions located in Denmark. The IT-architect brought his concerns to the management group where he argued for a reorganization of the entire project. The Danish IT-architect argued strongly against having the Philippine department do this task alone. The IT-architect told us that he had learned from previous experience that the Philippine office would not be able to handle this task single-handedly. While asked about the details regarding the contract the IT-architect said:

Actually I had no part in the original contract description. So when I finally get to see the contract together with a couple of colleagues, I realize how severe the situation is. I have at least four or five years of experience working with the Philippine office and I know fairly well of their strengths and weaknesses. The

requirement specification was clearly outside of their competences. Maybe not all of it but some of the components planned to be developed there. Which is why one of the first decisions we took was to withdraw parts of the project. That was not a very popular decision, but we did it.

It should be noted that at this point in time the Danes generally claimed that a successful offshore project in the company was yet to be seen and it was thus unrealistic that the Philippine office would be able to handle such a big project alone primarily because they were not technically equipped for the task or had the necessary experience. The Philippine employees had counter-argued by saying that they had the competences and the main problem was actually not lack of competences but rather how the collaborative work was done. Nevertheless, since this IT-architect was also very experienced and had worked for the company more than ten years his arguments carried much weight and the management group decided to follow his recommendations. His arguments against letting the Philippine office complete the project alone was weighed against the economic setback and the management group was convinced that the project had been scoped wrongly from the beginning. It was decided that Danish developers should develop approximately 50 % of the project. At this stage the organizational practice still largely dictated a practice of reducing links between the Danish and Philippine office thus moving everything offshore does not emphasize closely coupled work but a tendency to insist on the distinction of sites rather than collaboration.

4.2.2 Dividing the tasks across different sites

The decision to withdraw 50 % of the project from the Philippine office would seem to afford more links between the two offices but the initial work practices still focused on reducing links between the Danish and the Philippine employees instead of taking advantage of the diversity of competences across the two sites. The new scope of the project required the involvement of more Danish employees and the Danish IT-architect decided that the project should be divided into 4 separate releases. Release A was to be developed in Denmark because it was deemed to very complex both due to its technical nature and due to the required integrations with a range of systems operated by local Danish municipalities and institutions. Release B and C were estimated to be the least complicated in terms of the technical expertise required, partly because the sensitive personal data was all to be handled in Release A. The sensitive data was subject to legal requirements according to Danish law and was thus easier to handle in Denmark. Thus the Philippine office was assigned to develop Release B and C while the Danish office began development on Release A. The final release, which was called release D was left undecided at the time, but the Danish office, would most likely be in charge of the development. The work practices at this stage focused on limited links with few people “bridging” the two offices. A Danish IT-architect was the primary collaboration partner with Philippine office while the remaining Danish employees focused their effort on release A. The collaborative work consisted of weekly technical meetings with a video link. The Danish IT-architect would be present as well as the Philippine system analyst. Typically there would also be 2- 3 Philippine developers present and sometimes even the Philippine IT-architect. These meetings were used to discuss the finished software components, solve technical issues and discuss the development for the following week. The interaction can be characterized as closely coupled but it only reflected a relatively small group of people in the entire project.

Another example of the loosely coupled work at the time was happening between two groups of Danish employees. One group consisted of the IT-architects and developers who primarily worked with the Philippine office but they also engaged with group of employees from another department in the company, namely the back office. Back office was responsible for the implementation of the technical solutions as well as integrating the software with the new hardware required to handle the sensitive information. The two groups were not collocated to begin with and did not even share the same building and the individual dependencies on each other were poorly communicated. The back office group were not aware of their important role until late into the project and when they finally began working on the task they realized that the hardware was much more complex than anticipated. Although these two groups shared the same cultural background they lacked an understanding of the dependencies between them. Although the geographical distance was much shorter between the two Danish groups they also experienced a feeling of “us and them” in the work practices at this stage.

4.2.3 Few bridges in the collaborative work

Although the organizational practices from the first stage had been removed there was still a prevalent feeling of them and us in the collaboration both between Danes and Philippine employees but also between employees located in Denmark. The strict division of work with few links between the groups replaced the previous constraints of the fixed price model and travelling. Moreover even with the removal of the restrictions on travels still only one person had actually travelled to the Philippines in the first seven months of the project. During the following year another three Danish employees went to the Philippines for a period of one week. No Philippine developers visited the Danish office in the first two years of the project. The project consisted of up to forty people at peak times, which gives an indication of the low travel frequency during the initial stages of the project. The reasons given for this were firstly that few Danish employees were actually interested in leaving Denmark for 2-3 weeks because of their families and although the Filipino developers expressed their willingness to travel this opportunity were not utilized at this stage. Therefore much of the collaborative work was coordinated mainly through the requirement specification or the product descriptions that described the task. While this is a typical way of coordination offshore software development projects the complexity and scope of this particular project created closely coupled dependencies that were not easily communicated in documents. The Danish employees spent a lot of time writing these documents and when the descriptions were misunderstood it led to frustrations on both sides and costly extra hours fixing the errors. An IT-architect described the process like this:

I had hoped that it [the documents] had had a greater effect compared to the energy that I spent producing them. On many occasions I had to remind them [the Philippine developers] to read the documents thoroughly. I feel like they should have gotten the information if they had read the document. [...] It seems to me that everything is going a little bit too fast in the Philippines.

The loosely coupled work practices gave little opportunity for interaction. Employees knew little of each other across the geographical distance and social relations between the two offices were sparse or non-existent. Lack of interaction was not only prevalent across the geographical distance but also between the back office group and the developers located in Denmark. The loosely coupled work practices with the back office group lacked

coordination since the two groups were practically unaware of their individual dependencies and how to solve them. At this stage both the scope and the budget had exceeded the project plan and the situation was slowly becoming critical in the project.

4.3 The state of the project becomes critical

Eight months into the project serious problems began to emerge. The quality of release B did live up to the expected quality. The IT-architects in Denmark were frustrated because they had tried to anticipate these problems by being extra careful with the wording in the requirements specification to ensure that misunderstandings were minimized. Since their hourly rate was high the time was very costly for the project when it had not worked as intended. However few of the Danish employees questioned the method of communication. Instead they emphasized all the work that had gone into detailing the requirements specification – work that seemed useless now that the misunderstandings occurred. The Filipino employees questioned the communication format and argued that there should be a lot more emphasis on domain knowledge, since the software was going to be applied to a Danish context with very specific institutional structures. Critical information was communicated mostly by emails since only few people held video meetings across the two sites.

4.3.1 Changing to daily Scrum meetings

The developers in the Philippines said that the reason behind the poor quality was because they were implementing a new work practices. The Philippine project leader had adopted in the Philippine development team. This meant that the Philippine developers would now have daily sprint meetings and work in 2-weeks cycles instead of the 3-week cycle they used before. Also they changed the name of these cycles from iterations to sprints. The time spent learning the new procedure had caused a lower than expected result on Release B. A newly assigned Filipino project leader described the situation at the time like this:

[...] When I came into the project this was already in Scrum. Although I had my reservations initially because of the fixed project and the fixed timeline project. It seemed strange at the time doing scrum. But on the other side, it did help quite a bit. Maybe not so much in term of the deadlines, but in terms of actually, let's say having better deliveries, better quality. It did help. And we're reaping the fruits of that now for [Release B].

Adapting to the Scrum had been weird in the beginning but the change became rewarding for the developers in Philippines. During the same period the Danish project leader travelled to the Philippines to ensure that they were actually able to finish release C due to the low quality of Release B. At the time the project leader said that it had been close to a termination of the collaboration with the Philippine office after seeing the quality of Release B. However, after going to the Philippines and meeting with the developer team, he was convinced to continue the collaborative work. Once implemented the scrum meetings greatly helped the Philippine employees primarily because the Philippine development team would now meet every morning and discuss their current tasks. This helped them to see the overall picture of the individual tasks and delegate tasks if needed.

4.3.2 Formalizing communication patterns

Collaborative work practices changed slowly towards a more closely coupled practice in various ways after the crisis with Release B. One example of this was that the project members met face-to-face before the development of release C began in the Philippines. Two IT-architects travelled to the Philippines to kick-start the development of Release C. They held presentations for

the Philippines developers and one of the IT-architects held 3-hour long meetings with the system analyst to discuss the technical aspects and implications of Release C. Another example of closely coupled work practices that emerged had its course over a year of the project. A Danish IT-architect slowly also began to formalize communication structures between the developers to overcome the misunderstandings. He had been in charge of facilitating the development of Release B and was one of the few who frequently interacted with Filipino developers. The antecedent to formalizing the communication was a more flexible communication strategy that had been used from the beginning of the project. The flexible approach meant that the Philippine developers could direct all their queries and questions to the Danish IT architect. However, this eventually led to a situation where the Danish IT architect was spending most of his time answering questions from Philippine developers leaving him little time to do other tasks. The flexible communication strategy became too time consuming for the IT architect, so the team decided to try another approach:

We actually changed the model so that Tom (a Philippine system analyst) would be the primary contact person, and he would gain the knowledge discussing with me what should be developed, and then he will distribute the knowledge from there (Obs.: May 24th Denmark).

Thus, to reestablish a formalized communication pattern and create a less ambiguous process, the Danish IT architect devised a communication model based on his experience with the different communication norms across Denmark and the Philippines. The purpose of the model was to formalize the communication flow between the Danish IT architect and the Philippine developers. The process of establishing a shared communication pattern lasted more than a year, and during this period they experimented with different approaches for how to communicate across geographical and temporal distance. Formalizing the communication pattern was a non-linear process that emerged over time as the project members experimented with different approaches. The end result created more visibility and accountability in the communication between the developers. Instead of having a loosely coupled structure they established a pattern of shared dependencies in the flow of communication as they moved closer to an offshore collaboration.

4.3.3 Evaluating the collaborative work

Towards the end of a development cycle in the Philippine office (Release C) the Danish project leader held two separate evaluation sessions of the current status. The Danish meeting lasted more than three hours with 20 participants from the Danish office. Most surprisingly was the fact that the collaborative work with the Philippine office was only mentioned once during the meeting. Instead people discussed a range of challenges that the Danish developers had experienced during the first year of the project. One of the main concerns was the lack of coordination between two offices in Denmark, namely the developers and the back office people. The back office people had not been prepared for the tasks expected of them and they got involved in the project at a time, which was considered too late by many of the developers. Moreover people were not located in the same office space, which created a disconnection between the two groups. The IT-developers did not know who to talk to in the back office department and the back office were not prepared for the task in terms of the time required to solve the complex tasks. Thus “us/them” issues had been a big issue for the Danish developers, but unexpectedly “us/them” feelings emerged among Danish

colleagues and not across geographical distance in distributed collaboration.

The evaluation session with the Philippine office was held as a video meeting with five attendees. The main concern at this meeting was the status of the collaborative work between Denmark and the Philippines. Although it was recognized that the collaboration had struggled during release B it was agreed by all participants that the collaboration was much better now. They even stated that the collaboration had been the best to date in their experience.

4.4 Emerging closely coupled work practices

Four months later a range of events in the project established the foundation for close collaborative work. The catalyst for these changes was the critical status of the project at the time. The project was severely delayed due to unforeseen challenges with the technical solutions as well as the complexity of the task. Moreover costs had accelerated due to these delays and the client had fined the project for not fulfilling the obligations in the contract. This situation called for changes in the organization of the work. A group of managers began to closely monitor the project mainly to ensure that no further delays would occur in the project. Thus management applied a key constraint of limiting the time available. Daily board meetings were established in the Danish office where key employees from the project participated. The purpose of these meetings was to enable the managers to react to current or future challenges and consequently the management group closely monitored the project members. The typical setup of these meetings was a stand-up meeting that lasted about an hour where the participants discussed the status of the project. A manager would lead the discussion and keep people updated by writing key points on a white board. These meetings established visibility across the project areas and made people accountable for tasks. As the Danish project leader told us during an interview:

Being more integrated and having meetings often is a good idea in my opinion. The ordinary project leader meetings, where the project leader I drawing on a whiteboard and describes the status are not working as intended. Instead let us call it board meetings or stand-up meetings or whatever. These [meetings] are much more rewarding. We sit down together and address the issues when they occur. I think this is the reason why we succeeded – that constantly were ready to react to emerging issues.

Urgent challenges could be addressed directly by several members of the team and the access to managers helped the project because they enabled a flow of resources. One example of this was an update to the software used in the Philippine office or improving the bandwidth. The presence of the managers shortened the decision time and increased the speed of which decisions were taken.

4.4.1 Daily test meetings

Another example of the transition towards more closely work practices was the introduction of daily test meetings. These meetings had no agenda and the purpose was to create a common ground between the distributed testers and discuss current challenges in the test process. Although some of the testers felt it awkward to attend a meeting with no agenda they quickly got accustomed to the new format. The meeting served as an open space of interaction where information could flow seamlessly among the participants. For instance during one of the daily test meeting we observed the following:

They talk about test issues that has occurred since last meeting and pin point aspects which are need to be resolved to be able to continue. Diane mentions that she has been talking to John earlier today. Mary refers to her interaction with Mike while Sean refers to Jack about a task, which he must do on Release D, so that they can continue the work. However Diane informs her that Jack apparently is sick today, but may be working from home. She is unsure about this since received an email earlier today from him. (Obs. Denmark Oct 03. 2013)

Although Mary, Diane and Sean are the only people actively present at the meeting they still expand the flow of information to include members outside of the meeting by referring to other people's current status on different tasks. We also observed how these meeting allowed the testers to praise or critique each other directly and thus holding people accountable for their work. In the final stages of the project the Filipino testers became responsible for testing of Release D, which was the final release. It was rather remarkable event given the initial mistrust between the Danish and Philippine employees.

4.4.2 Project leader meetings

The critical situation and the involvement of the management level resulted in new emerging collaborative work practices that created more links between people in the project. At this point in time project leader meetings had been relatively rare and had rarely happened in the first year of the project. This procedure was changed when a new project leader was assigned in Denmark. The new project leader quickly initiated weekly project leader meetings with the Philippine counterpart. Later these meetings also included people from the development group as well as key managers. The project leader began to have weekly project leader meetings with the project leader in the Philippines. Interestingly the project leader in Denmark also invited the IT-architects and managers from both the Danish and the Philippine offices to these meetings. The key difference from ordinary project leader meetings was that people from different organizational levels were able to meet and discuss the current status of the project. Thus the IT-architects could explain the challenges they were currently facing and the managers would be given a chance to better understand the complexity of the work. Moreover the managers were able to make quick decisions that would normally have to be confirmed at the monthly steering meeting. By having people from different organizational levels the project leader facilitated an effective forum for solving critical problems. Information could travel seamlessly between different levels in the organization and also to other employees in the project through the participants at the meeting. At one of these meetings the project leader successfully argued for more collocation between the developers and the back office people in Denmark. The project leader told us later that:

I think that the main factor [for succeeding] was allowing us to be collocated. I mean having the project members [in Denmark] sitting together and use the resources from the back office that was required. This has been really important for the project.

Being collocated greatly alleviated the disconnection between the developers in Denmark and the back office people. The project leader meeting established an easier access to resources and a shared understanding of the complexity and what needed to be done to solve challenges – both in Denmark and in the Philippines.

5. DISCUSSION

Why did the IT-developers shift towards more closely coupled work? First it became evident in the data analysis that the company initially still treated the Philippine office as an outsourcing capacity for example in terms of sticking to the fixed price setup despite being fully integrated into the same company. The collaborative aspect of the work was minimized and rather seemed to function as two separate entities that were merely solving a task together with little mutual responsibility. The two offices kept working as separate units even when after the fixed price setup had been abandoned. For example illustrated by the single point of contact between the two offices, which allowed for very little personal contact across the two teams. Another example of the separate nature of the work was the fact that the Philippine office implemented Scrum in the development process without the Danes even being aware of this change. Work practices were loosely coupled in terms little contact between the teams for example illustrated by the strict division of tasks with only a single-point of contact. Tasks were divided or decomposed into smaller pieces with the intention of minimizing interaction. The strategy of dividing tasks is typical in the software development business (Grinter 2003) and reducing interaction have been proposed as a viable strategy in global software development projects (Hertzum and Pries-Heje 2011). However we saw that in our case the separation of work practices subsequently created a perception of two separate entities. The loosely coupled work practices established a difference between how employees viewed themselves as one entity and then how they viewed their remote colleagues as another entity.

5.1 Alleviating the feeling of “us and them”

This distinction between ‘us’ and ‘them’ eliminated the possibility of perceiving the project as one system conducted with one team who just happens to be located at two different geographic sites. Instead the project seemed to consist of two teams with few shared relations and who had separate responsibility for clearly defined parts of the same system. Prior research has investigated the notion of “us and them” in terms of faultlines e.g. when key attributes of one group correlates without cross-cutting ties between other groups (Cramton and Hinds 2005). In our case we observed how the IT developers in the project initially shared few aligning attributes or were unaware of their existence. The developers located at the Danish office pointed to all the work they had put into the requirement specification and the lack of experience in the Philippine office as the reason for the poor quality of Release B. On the other hand the developers in Philippine office pointed to communication practices of the Danish office, which did not put enough emphasis on critical domain knowledge. Interestingly the IT-developers in Denmark also felt isolated from the Danish back office although they shared same cultural background and geographical location. Consequently the feeling of acting like “us and them” was an issue both between Denmark and the Philippines as well as between two groups of Danish employees located at the same site.

Release B, developed in the Philippines, did not meet contractual expectations. Failed expectations and lack of social interaction can result in lack of trust in virtual teams (Jarvenpaa and Leidner 1999) which also was apparent in this project. Interestingly the IT-developers managed to overcome the critical situation by choosing to engage in more closely coupled work. One example of the shift towards more closely coupled practices was the intensification of travels between the two offices. Travels allowed more face-to-face time building commitment in the project (Nardi,

2005) and created insight into local context and conditions. For instance the Danish IT-developers expressed their surprise of how cold the Philippine office was due to the air conditioning causing some Philippine developers to wear clothes during office hours. They also experienced the bandwidth problems that their Filipino colleagues had to deal with every day. Similarly the Filipino developers who travelled to Denmark felt the nerve and stressful environment in the Danish office. The travels enabled the IT developers to recognize context specific work conditions in each group. The travels were the first steps towards more closely coupled work practices because the IT-developers spent time together sharing knowledge and getting to know each other. Another similar change occurred when the Danish project leader decided to collocate the Danish Back office with the Danish developers. The benefit of being collocated was that they could see each other and knew that they were all working on the same project. While the importance of face-to-face time in global software development has been discussed at length in the literature (Oshri et al. 2007, Herbsleb et al. 2005), we point to these travels and the collocation of Danish employees as a turning point towards more closely coupled work in collaboration because it enabled the IT-developers to create closer connections to each other and begin to establish shared social norms. For instance during one of the travels the IT-architect spent time with the Filipino developers discussing the definition of “done” and finally decided that “done” meant tested and ready for implementation. Moreover seeing each other at the work place established a foundation for social translucence as people and their activities became mutually visible. We saw that the IT-developers managed to alleviate the feeling of “us and them” by intensifying the collaboration instead of continuing with loosely coupled work practices. The collocation greatly helped the collaborative work between the two Danish groups.

5.2 Creating connections across the project

The shift towards more closely coupled work also alleviated other challenges in the project. Initially the developers had trouble understanding each other while only sharing few links for instance illustrated by the failure to meet expectations of Release B. Although tasks (or Releases) were distributed between the Danish and Philippine office the different software components were highly dependent on each other and required coordination between the developers. This is what Grinter (2003) refers to as the recomposing aspect of the software. In time the work practices changed in several ways to overcome the challenge of recomposing the software. The developer teams introduced daily scrum meetings in the Philippines and the IT-architect formalized the communication pattern between developers in the Philippines and in Denmark. The testers on both sides began to have daily meetings and the project leaders also engaged in weekly meetings. These changes were often not planned but came as a reaction to discrepant (Majchrzak et al. 2000) events such as unexpected technical complexities or people suddenly leaving the project. Interestingly the developers chose more closely coupled work as a reaction to the discrepant events. Working closely together created more connections in the team allowing easier knowledge exchange and shared responsibility of the outcome. For instance sharing knowledge by presenting the requirement specification for the next release not only in written documents but also in face-to-face presentations where both Danish and Filipino developers engaged in discussions. The importance of meeting face-to-face cannot be understated in terms of establishing commitment and common ground as recent research has also argued (Nardi 2005, Olson and Olson 2000). However, the daily contact for instance

enabled through video meetings also contributed to greatly improving the collaboration by enabling visibility and awareness between project members across the geographical distance. Research have pointed to the concept of social translucence as a key factor for collaboration (Erickson and Kellogg 2000). In a collocated setting project members are easily visible and they are mutually aware of each other. The team members felt more connected to each other and the flow of communication became clearer. One Filipino developer specifically said that the communication was less “cloudy” after the daily meetings were established. In our case the developers worked distributed but we argue that the shift towards more closely coupled work practices also enabled mutual visibility of each other. By making more connections between people in the project they gained insight into the different tasks people were working on and allowing information to travel more freely between distributed members. The example with the daily test meetings illustrates how daily meetings created mutual visibility across different members. The closely coupled work practices thus created more connections across the project both in terms of spanning the geographical distance and locally. These connections enabled information to flow freely between members and creating insight in individual challenges of the project members.

5.3 Creating connections across hierarchies

Lastly we argue that closely coupled work practices enabled more connections between the traditional organizational hierarchies. Daily board meetings and the new project leader meetings spanned IT-architects, system analysts, project leaders and managers. Normally the project leader would report back to the steering group consisting of managers and later report back to the project members. However, we have described how for instance a Filipino system analyst or a Danish IT-architect were able to provide the necessary descriptions of current technical complexities in the development to both the project leader and the managers present at these meetings. Having these people in the project leader meetings made it easier for both project leaders and managers to understand the challenges in the project and acknowledge that action was needed. Bridging the gap between the organizational hierarchy greatly benefitted the project in the later stages especially when constrains such as time and money made conditions difficult. The managers provided decision-power and were able to react quickly to current issues based on the information received directly from the IT-developers. One example of this could be quickly adding new resources to the project in terms of necessary software updates or increasing the bandwidth in the Philippine office. Consequently having the project leader meeting and the daily board meetings allowed quick reactions to challenges that could be acknowledged by all parties at the meeting. Creating closer connections between the management group and the project members also increased the flow of information between people that would normally not work closely together. The close collaboration across hierarchies helped greatly in the later stages of the project when time became a critical factor. The meetings facilitated shared information between practitioners and managers and ensured that direct action would address the most problematic issues.

6. CONCLUSION

This paper presents an empirical observation from a longitudinal ethnographic study of a global software development company demonstrating why IT-developers chose to shift from loosely coupled to closely coupled work. We found that the IT-developers chose to engage in close collaboration to solve a complex and

highly interdependent task while assessing the collaborative work as the best to date. We propose three key findings of why closely coupled work improved the collaborative practices in a geographically distributed setting. Firstly, working closely coupled work practices established many connections across the collaboration ensuring knowledge exchange and improving coordination for instance by establishing daily test meetings, collocating people and formalizing communication procedures between developers. Secondly, closely coupled work practices diminished the formation of sub-groups locally and established new faultlines across the geographical distance by removing the organizational constraints of the fixed price model and intensifying travels between the two locations. By collocating Danish employees and increasing the frequency of meetings between distributed members the IT-developers managed to moderate patterns of “us” and “them” in both locally and remotely in the company. Finally, closely coupled work created connections across organizational hierarchies by introducing daily board meetings and new project leader meetings. These meetings increased awareness of the complexities in the project, made people accountable and enabled shared responsibility the project outcome. Moreover the connections across organizational hierarchies allowed information to travel seamlessly between layers in the organization and consequently the practitioners could better anticipate issues and act accordingly for instance by adding more resources to the project. While it is impossible to generalize based on a single work place study, the data material showed that closely coupled work practices enabled productive collaboration in this particular project. This suggests that closely coupled work practices are useful for small to medium sized teams that collaborates over extended periods of time and operates in a field that requires specific domain knowledge. Further research is needed to investigate how closely coupled work practices scales to larger teams as well as how closely coupled work supports teams for shorter periods of time and in less domain specific projects.

In conclusion closely coupled work aided the IT-developers in solving the task despite being very challenging and suffering from delays and economic loss. Not only did they solve the task they also experienced that the collaborative work as the most successful to date. Thus transforming work practices from loosely coupled to closely coupled practices became a valuable learning process for doing global software development in the company.

7. ACKNOWLEDGMENTS

This research has been funded by the Danish Agency for Science, Technology and Innovation under the project "Next Generation Technology for Global Software Development", #10-092313. We would like to thank our study participants who generously spent time with us and shared their thoughts. We are grateful to anonymous reviewers for a close reading of the manuscript and their helpful suggestions.

8. REFERENCES

- [1] Avram, G., Bannon, L., Bowers, J., Sheechan, A. and Sullivan, D. K. (2009) 'Bridging, Patching and Keeping the Work Flowing: Defect Resolution in Distributed Software Development', *Computer Supported Cooperative Work (CSCW)*, 18, 477 - 507.
- [2] Balakrishnan, A. D., Kiesler, S., Cummings, J. N., & Zadeh, R. (2011). 'Research Team Integration: What It Is and Why It Matters' (p. 523). in *Conference on Computer Supported Cooperative Work (CSCW)*, New York, New York, USA. ACM Press, 523 -532.
- [3] Bhat, J. M., Gupta, M. and Murthy, S. M. (2006) 'Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing', *IEEE Software*, 23(5), 38 - 44.
- [4] Bjørn, P. and Ngwenyama, O. (2009) 'Virtual Team Collaboration: Building Shared Meaning, Resolving Breakdowns and Creating Translucence', *Information Systems Journal*, 19(3), 227 - 253.
- [5] Boden, A., Avram, G., Bannon, L., & Wulf, V. (2012). 'Knowledge Sharing Practices and the Impact of Cultural Factors: Reflections on Two Case Studies of Offshoring in SME', *Journal of Software Evolution and Process*, 24(2), 139 - 152.
- [6] Boden, A. and Avram, G. (2009) 'Knowledge Management in Distributed Software Development Teams: Does Culture Matter?', in *International Conference on Global Software Engineering*, Limerick, Ireland, IEEE Press, 18 - 27.
- [7] Boden, A., Nett, B. and Wulf, V. (2009) 'Trust and Social Capital: Revisiting an Offshoring Failure Story of a Small German Software Company', in Wagner, I., Tellioglu, H., Balka, E., Simone, C. and Ciolfi, L., eds., *European Conference on Computer Supported Cooperative Work (ECSCW)*, Vienna, Austria, Springer, 123 - 142.
- [8] Christensen, L. R. and Bjørn, P. (2014) 'Documentscape: Intertextuality, Sequentiality & Autonomy at Work', in *ACM CHI Conference on Human Factors in Computing Systems*, Toronto, Canada, ACM Press,
- [9] Cramton, C. D. and Hinds, P. (2005) 'Subgroup Dynamics Internationally Distributed Teams. Ethnocentrism or Cross-National Learning?', *Research in Organizational Behavior*, 26, 233 - 265.
- [10] Eisenhardt, K. M. (1989) 'Building Theories From Case Study Research', *The Academy of Management Review*, 14(4), 532 - 550.
- [11] Erickson, T. and Kellogg, W. A. (2000) 'Social Translucence: An Approach to Designing Systems that Support Social Processes', *ACM Transactions on Computer-Human Interaction*, 7(1), 59 - 83.
- [12] Gerson, E. M. (2008) 'Reach, Bracket, and the Limits of Rationalized Coordination: Some Challenges for CSCW' in Ackerman, M. S., Halverson, C. A., Erickson, T. and Kellogg, W. A., eds., *Resources, Co-Evolution and Artifacts. Theory in CSCW*, Springer 193 - 220.
- [13] Grinter, R. E. (2003) 'Recomposition: Coordinating a Web of Software Dependencies', *Journal of Computer Supported Cooperative Work*, 12, 297 - 327.
- [14] Heath, C. and Luff, P. (1992) 'Collaboration and Control: Crisis Management and Multimedia Technology in London Underground Line Control Rooms', *Journal of Computer Supported Cooperative Work*, 1(1), 24 - 48.

- [15] Herbsleb, J., Paulish, D. and Bass, M. (2005) 'Global Software Development at Siemens: Experience from Nine Projects', in ICSE, St. Louis, Missouri, USA.,
- [16] Herbsleb, J. D. (2007) 'Global Software Engineering: The Future of Socio-technical Coordination', in Future of Software Engineering, IEEE, 188 - 198.
- [17] Herbsleb, J. D., Mockus, A., Finholt, T. A. and Grinter, R. E. (2000) 'Distance, Dependencies and Delay in a Global Collaboration', in Conference on Computer Supported Cooperative Work (CSCW), Philadelphia, Pennsylvania ACM Press, 319 - 328.
- [18] Hertzum, M. and Pries-Heje, J. (2011) 'Is Minimizing Interaction a Solution to Cultural and Maturity Inequality in Offshore Outsourcing?' in Hertzum, M. and Jørgensen, C., eds., *Balancing Sourcing and Innovation Systems Development*, Trondheim, Norway: Tapir Academic Press, 77 - 97.
- [19] Hofstede, G., & Hofstede, G. J. (2004). *Cultures and Organizations: Software for the Mind*. United States, McGraw-Hill Professional.
- [20] Jarvenpaa, S. and Leidner, D. (1999) 'Communication and Trust in Global Virtual Teams', *Organization Science*, 10(6), 791 - 815.
- [21] Jensen, R. E. and Bjørn, P. (2012) 'Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds', in 10th International Conference on the Design of Cooperative Systems (COOP), Marseilles, France, Springer, 123 - 136.
- [22] Krishna, S., Sahay, S. and Walsham, G. (2004) 'Communications of the ACM', *International Conference on Intercultural Collaboration*, 47(4), 62 - 66.
- [23] Majchrzak, A., Rice, R. E., Malhotra, A., King, N. and Ba, S. (2000) 'Technology Adaption: The Case of a Computer-Supported Inter-Organizational Virtual Team', *MIS Quarterly*, 24(4), 569-600.
- [24] Mark, G. (2002) 'Conventions and Commitments in Distributed CSCW Groups', *Journal of Computer Supported Cooperative Work (CSCW)*, 11(3), 349 - 387.
- [25] Marrewijk, A. v. (2010) 'Situational Construction of Dutch-Indian Cultural Differences in Global IT Projects', *Scandinavian Journal of Management*, 26(4), 368 - 380.
- [26] Matthiessen, S., Bjørn, P. and Petersen, L. M. (2014) "'Figure Out How to Code with the Hands of Others": Recognizing Cultural Blind Spots in Global Software Development', in Conference on Computer Supported Cooperative Work (CSCW), Baltimore, USA, ACM Press,
- [27] Metiu, A. (2006) 'Owning the Code: Status Closure in Distributed Groups', *Organization Science*, 17(4), 418 - 435.
- [28] Mockus, A. and Weiss, D. M. (2001) 'Globalization by Chunking: A Quantitative Approach', *IEEE Software*, 18(2), 30 - 37.
- [29] Nardi, B. (2005) 'Beyond Bandwidth: Dimensions of Connection in Interpersonal Communication', *Journal of Computer Supported Cooperative Work (CSCW)*, 14, 91 - 130.
- [30] Neale, D. C., Carroll, J. M. and Rosson, M. B. (2004) 'Evaluating Computer-Supported Cooperative Work: Models and Frameworks', in Conference on Computer Supported Cooperative Work (CSCW), Chicago, Illinois, ACM Press, 112 - 121.
- [31] Noll, J., Beecham, S. and Richardson, I. (2010) 'Global Software Development and Collaboration: Barriers and Solutions', *ACM Inroads*, 66 - 78.
- [32] Olson, G. M. and Olson, J. S. (2000) 'Distance Matters', *Human-Computer Interaction*, 15, 139 - 178.
- [33] Oshri, I., Kotlarsky, J. and Willcocks, L. P. (2007) 'Global Software Development: Exploring Socialization and Face-to-Face Meetings in Distributed Strategic Projects', *Journal of Strategic Information Systems*, 16, 25 - 49.
- [34] Pinelle, D. and Gutwin, C. (2003) 'Designing for Loose Coupling in Mobile Groups', in Conference on Supporting Group Work (GROUP), Sanibel Island, Florida, ACM Press, 75 - 84.
- [35] Schmidt, K. (2011). *Cooperative Work and Coordinative Practices*. Springer Science & Business Media.
- [36] Schmidt, K. (2002) 'The Problem with 'Awareness'', *Journal of Computer Supported Cooperative Work*, 11, 285 - 298.
- [37] Schmidt, K. and Bannon, L. (1992) 'Taking CSCW Seriously: Supporting Articulation Work', *Journal of Computer Supported Cooperative Work (CSCW)*, 1(1 - 2), 7 - 40.
- [38] Søderberg, A.-M. and Holden, N. (2002) 'Rethinking Cross Cultural Management in a Globalizing Business World', *International Journal of Cross Cultural Management*, 2(1), 103 - 121.
- [39] Søderberg, A.-M., Krishna, S. and Bjørn, P. (2013) 'Commitment, Trust and Cultural Sensitivity in Strategic Partnerships', *Journal of International Management*, 19(4), 347 - 361.
- [40] Strauss, A. and Glaser, B. G. (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research*, United States: Rutgers.