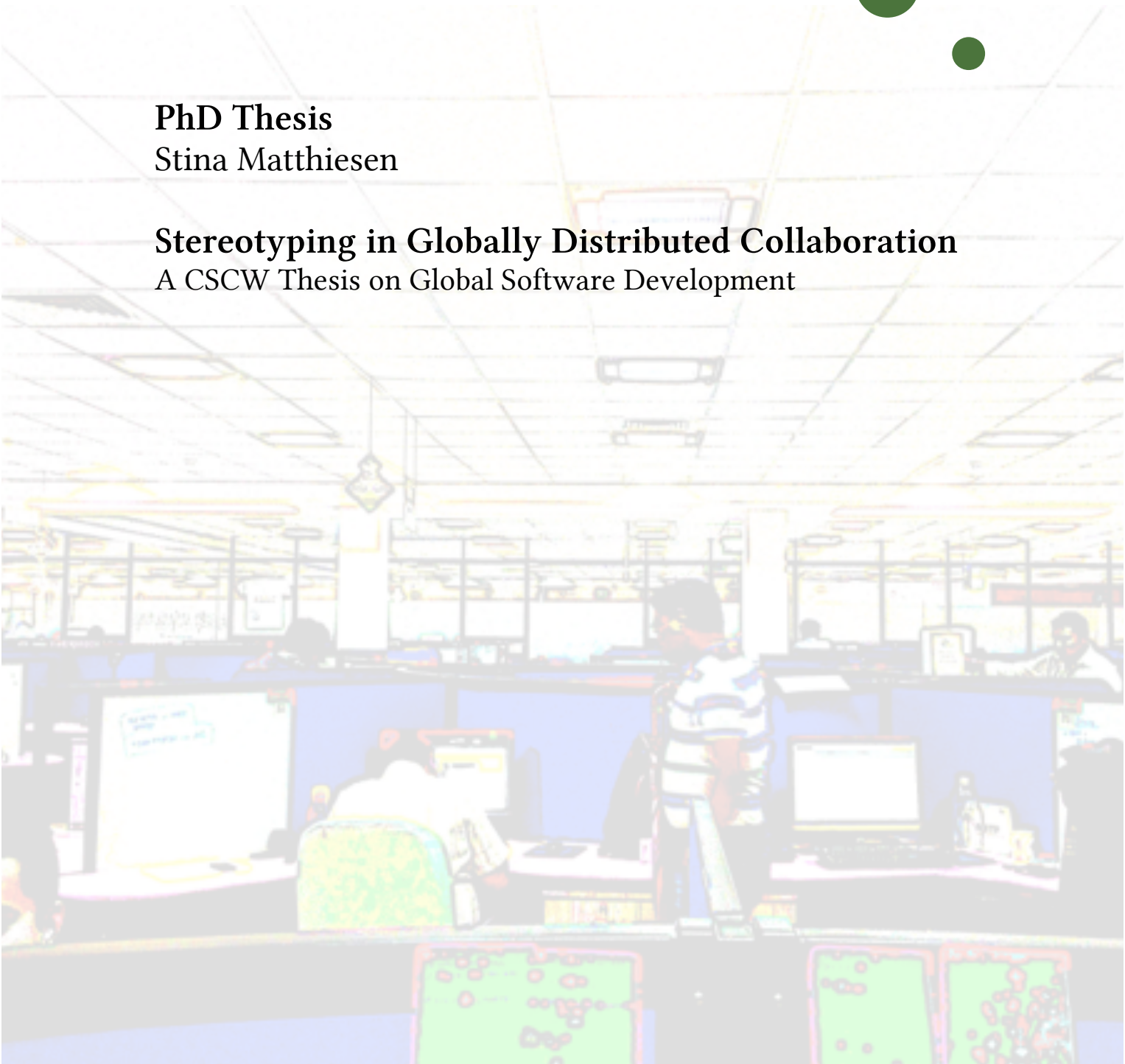
UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE



PhD Thesis

Stina Matthiesen

Stereotyping in Globally Distributed Collaboration

A CSCW Thesis on Global Software Development

Submitted
July 19th, 2019

Academic Advisor
Pernille Bjørn, Professor, University of Copenhagen

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen

Colophon

Author

Stina Matthiesen
Human-Centered Computing (HCC)
Department of Computer Science (DIKU)
University of Copenhagen

Title of PhD Thesis

Stereotyping in Globally Distributed Collaboration
A CSCW Thesis on Global Software Development

Submitted/Defended

July 19th 2019/TBA

Academic Advisor

Pernille Bjørn, Professor, University of Copenhagen

Assessment Committee

Yvonne Dittrich, Associate Professor, IT University of Copenhagen
Liam Bannon, Professor Emeritus, University of Limerick
Finn Kensing, Professor, University of Copenhagen

Cover Photo
Fieldwork photo (edited) from India, empirical study no. 1

PART I

Acknowledgements

This PhD project has been a long and exciting journey. It has also been a journey I would not have embarked upon or indeed finished if it had not been for all the amazing people who have supported me along the way.

First and foremost, I want to say a heartfelt "thank you" to my supervisor, Pernille Bjørn, for providing continuous and cheerful support beyond anything I could expect from a supervisor, and for helpfully deconstructing my work again and again, and guiding me in recomposing it into great work. You taught me that a rough draft was better than no draft at all for discussing where my research was heading. You have been a tremendous help and offered me ambitious and exciting scholarly advice. As if that is not enough, I would also like to thank you for your endless patience with me when plans change as life happens, daughters are born, and a family is rejoined in Korea. You are a true inspiration, both as a scholar and as a caring human being.

Wholehearted thanks are also due to all the anonymous informants in the companies I have studied. Without you welcoming me into your world and sharing your thoughts, lunches, and work with me, this project would not exist. Thanks for answering my endless questions and letting me study all of you. I know that it has, for some of you, been out of your comfort zone, as it sometimes has for me too. Thanks for making me feel part of the team, for inviting me to your homes, and for making fun of me with my fellow companion: my little black notebook. This project would not have been as fun without you all.

Thank you to all the amazing people affiliated with the Next Generation Technologies and Processes for Global Software Development (NeXGSD) project, and big thanks to the people from the Computer-Supported Cooperative Work (CSCW) community: Ingrid Erickson, Dave Randall, Jørgen Bansler, Marissa Cohn, Paul Dourish, Jackie O'Neill, and numerous others who have kindly offered their time, scholarly advice, and comments, which have helped me form the project into something I am proud of.

Acknowledgements

Both at the IT University of Copenhagen (ITU) and the Department of Computer Science at University of Copenhagen (DIKU) I have found inspiring and friendly people who have made going to work a joy. Thank you to Tariq Osman Andersen, Nelson Tenóri, Jarrod Knibbe, Joanna Bergström-Lehtovirta, Xiaoyi Wang, Aske Mottelson, Henning Pohl, Kasper Hornbæk, and Jeppe Eimose Waagstein. Also a special thanks to fellow PhD students who—at some point in time—accompanied me along the way, Nanna Gorm, Karim Jabbar, and Rasmus Eskild Jensen. To María Menendez-Blanco, an amazing colleague and friend, thanks for all the honest and kindhearted conversations we have had, and the many procrastination coffees we have shared. Naja Holten-Møller, thanks for your endless support of my work and you our *søster sind*. I treasure your insightful and sometimes thought-provoking comments, and of course our friendship. To Claus Trillingsgaard, thanks for letting me into your world, for granting me access to exciting empirical sites, and for supporting and commenting on my work along the way.

Thanks to my parents for helping out our family whenever a deadline, conference, doctoral colloquium, or summer school needed my full attention. I somehow love how you keep referring to my work as "homework," making me much younger than my birth certificate states. Thanks to my dear friends: Julie, Edith, Signe, Nanna, Lise, Martina, and several more, for taking my mind off the project when it was most needed, and for listening and caring whenever I needed to talk about work.

The biggest gratitude goes to my best friend and partner in life, Kristoffer, for your love, support, and patience. You have been with me from the very beginning and have listened to all my complaints, joys, and indecisive deliberations along the way. Thank you for that Stof; I love you so much. Starting a family with you during my PhD project is without a doubt the most beautiful and important achievement we have accomplished and I feel grateful that we were able to, even though the PhD project was stretched by a few years. To my two wonderful girls, Vera Mi and Esther Joo, thanks for teaching me things about myself I did not know before and for motivating me to the finish the dissertation. I hope you know I love you dearly.

Stina Matthiesen
July 2019

Summary

This PhD dissertation investigates the encounters that take place when IT professionals participate in global software development (GSD) work at multiple locations. The dissertation explores the coordination, communication, and use of cooperative technologies, which IT professionals engage in when organizing their individual yet interdependent work tasks. The empirical foundation of the dissertation comprises three multi-sited ethnographic studies of GSD practices in three Danish IT companies and their global IT vendors in India or Poland. Large-scale software systems are not easy to develop in global setups, and IT professionals face a range of challenges on a daily basis. Early on in this doctoral research, *cultural differences* served as the most frequent answer when the IT professionals were asked about the reasons for their daily challenges. The everyday language expressed by the IT professionals situated in Denmark often involved descriptions of negative national cultural stereotypes pertaining to the remote collaborators situated in India or Poland. Yet, through closer investigation, it was clear that the challenges merely related to coordination and communication issues. In this way, the vocabulary that is currently in use is insufficient to fully capture and address the actual challenges that exist in GSD. Nevertheless, the use of negative stereotyping in GSD is rarely addressed by corporations or in research within the domain.

This dissertation extends the set of theoretical concepts that guides us as we think through how to address the negative stereotyping and support GSD work. It does so by offering contributions to Computer-Supported Cooperative Work (CSCW) by exploring the following research question: *How can researchers and IT professionals move beyond negative stereotyping and instead address the concrete coordination and communication practices that cause problems in global software development?*

The dissertation compiles findings from a collection of four published papers and proposes new the ways to address negative stereotyping in GSD. As a novel contribution to CSCW, the concept of *implicit bias* is introduced to help explain why even the most well-intentioned people draw on stereotypes and apply prejudiced descriptions of their foreign and remote colleagues when encountering issues in GSD. Now, to move beyond negative stereotyping IT professionals must pay

attention to the various ways that ***categories***, ***power hierarchies***, ***misconceptions***, and ***implicit biases*** contribute to the explanation of collaborative challenges. By doing so, IT professionals will be able to understand the mechanisms behind what implicitly influences the way we organize and act in the world; they will be able to uncover specific collaboration-related problems in richer detail. The dissertation offers a **GSD Stereotype Framework** that describes three main areas that IT professionals and researchers should examine closely to detect problems in GSD:

1. ***The organization of work.*** This requires analytical investigations that take into account the particular way implicit biases and power hierarchies influence the way work tasks are divided and distributed, or how efforts to communicate are interpreted as work or interruptions.
2. ***The collaborative technologies and system structures.*** These include considerations on the categories embedded in the collaborative tools and systems as they have the ability to hide existing issues or discrepancies in the collaboration. Additionally, the power hierarchies and implicit biases deserve attention as these are expressed along with or through the categorization schemes that underlie the tools and system structures applied in GSD.
3. ***The conditions for work.*** To understand these, attention must be paid to the infrastructural limitations or physical constraints for participating in GSD work. Here, the implicit biases and misconceptions that collaborators may have about working at a certain remote location are important analytical focuses when distributed collaborators' work is misinterpreted and explained through negative stereotypes.

By introducing the GSD Stereotype Framework to the empirical field, the dissertation ends by offering insights into how best to address negative stereotyping in practice. It encourages collaborators to avoid reducing collaboration issues to differences in national culture. Discussing implicit bias in GSD shows that people are not only supported but also inspired to move discussions from simplistic stereotypical explanations to in-depth considerations on the underlying practices, systems, and conditions that enable or constrain GSD work. The contributions set the stage for further investigations on how to intervene in practice and, in particular, on how to bring concepts such as implicit bias to attention as a complementary gaze on globally distributed collaboration.

Sammenfatning

Denne ph.d.-afhandling undersøger de samarbejdsrelaterede problemer, som finder sted når it-fagfolk arbejder fra flere forskellige lokationer i global softwareudvikling (GSD). Specifikt undersøger afhandlingen den koordinering, kommunikation og anvendelse af samarbejdsteknologier, som it-fagfolk engagerer sig i, når de organiserer deres individuelle men indbyrdes afhængige arbejdsopgaver. Afhandlingens empiriske grundlag består af tre etnografiske undersøgelser af GSD i praksis - foretaget i tre danske it-virksomheder og ved deres globale it-leverandører i Indien eller Polen. Store komplekse it-systemer er ikke lette at udvikle gennem GSD, og it-fagfolk står overfor en række daglige udfordringer. Tidligt i denne forskning viste det sig, at kulturelle forskelle var det mest hyppige svar, når it-fagfolk blev spurgt om årsagerne til deres daglige udfordringer. Herunder involverede det daglige sprog (iblandt it-fagfolk i Danmark) ofte beskrivelser af negative, nationale kulturelle stereotyper af deres samarbejdspartnere i Indien eller Polen. Ved nærmere undersøgelse blev det dog klart, at udfordringerne primært handlede om problemer med koordinering og kommunikation. Således er det ordforråd, som i øjeblikket anvendes, utilstrækkeligt til at indfange og adressere de faktiske udfordringer, der finder sted i GSD. Ikke desto mindre bliver brugen af kultur og negative stereotyper sjældent adresseret af virksomheder eller i forskning inden for domænet.

Denne afhandling adresserer ovenstående ved at udvide sættet af teoretiske koncepter, som kan hjælpe os med at navigere og tænke i, hvordan man kan håndtere negative stereotyper og understøtte samarbejdet i GSD. Således bidrager afhandlingen med viden til det forskningsperspektiv, der vedrører computer-understøttede samarbejdspraksisser (CSCW), ved at udforske følgende forskningsspørgsmål: *Hvordan kan forskere og it-fagfolk bevæge sig videre fra brugen af negative stereotyper og i stedet tage hånd om de konkrete koordinerings- og kommunikationspraksisser, der forårsager problemer i global softwareudvikling?* Ud fra resultater fra fire publicerede artikler foreslås nye måder at håndtere brugen af negativ stereotyper i GSD. Som et nyt bidrag til CSCW introduceres begrebet *implicit bias* til at hjælpe med at forklare, hvorfor selv de mest velmenende mennesker trækker på fordomsfulde beskrivelser af deres udenlandske kolleger, når

de støder på problemer i GSD. Specifikt beskriver afhandlingen, hvordan it-fagfolk såvel som forskere skal være opmærksomme på de forskellige måder, hvorpå **kategorier**, **magthierarkier**, **misforståelser** og ***implicit biases*** bidrager negativt til at forklare udfordringer i GSD. Ved således at forstå mekanismerne bag, hvad der implicit påvirker måden, vi organiserer og handler i verden, vil vi være i stand til at afdække de konkrete samarbejdsrelaterede problemer indenfor GSD.

Afhandlingen tilbyder et begrebsapparat, der beskriver tre hovedområder, som bør undersøges for at håndtere brugen af stereotyper i GSD: 1) **Arbejdets tilrettelæggelse**: Dette kræver analytiske undersøgelser, der tager særligt højde for, hvordan *implicit bias* og magthierarkier påvirker måden, hvorpå arbejdsopgaver opdeles og distribueres, eller hvordan indsatsen for at kommunikere fortolkes som reelt arbejde eller afbrydelser. **2) Samarbejdsteknologier og systemstrukturer**: Disse omfatter analytiske overvejelser omkring, hvordan kategorierne, der ligger til grund for de værktøjer og systemstrukturer anvendt in i GSD, kan skjule eksisterende problemer eller uoverensstemmelser i samarbejdet. Ligeledes er det vigtig at være opmærksom på, hvilke magthierarkier og *implicit biases,* der kommer til udtryk sammen med eller gennem de anvendte kategorier. **3) Arbejdsbetingelserne**: For at forstå disse skal der tages hensyn til de infrastrukturmæssige eller fysiske begrænsninger, der er i GSD. Når distribuerede medarbejderes arbejde misfortolkes og forklares gennem negative stereotyper, er det vigtigt at forstå, hvad det er for *implicit biases* og misforståelser, som samarbejdspartnere har omkring det arbejde, der udføres fra en bestemt geografisk placering.

Til slut introduceres ovennævnte begrebsapparat til det empiriske felt, hvilket giver indsigt i, hvordan man kan blive opmærksom på brugen af stereotyper i praksis og dermed opfordre samarbejdspartnere til at undgå at reducere samarbejdsproblemer til forskelle i national kultur. Ved at sætte fokus på *implicit bias* i GSD, viser afhandlingen, at folk kan støttes og inspireres til at flytte diskussioner fra forenklede forklaringer til dybdegående overvejelser omkring, hvordan de underliggende praksisser, systemer og betingelser muliggør eller begrænser arbejdet i GSD. På den måde sættes scenen for yderligere undersøgelser af, hvordan man kan gribe ind i praksis, og særligt hvordan man kan bringe et begreb som *implicit bias* i spil som en komplementær vinkel på globalt distribueret samarbejde.

Contents

1 Introduction

In today's globalized economy, it is the new normal for IT companies to have part of their software development work taken up by IT labor located in countries with lower wages (Walsham, 2002; Herbsleb, 2007). This kind of software development is broadly termed as *global software development (GSD)* and often it involves IT developers with different national and organizational cultures collaborating across different geographic locations and time zones, using various traditional and IT-enabled means to collaborate (Hossain et al., 2011). In this dissertation I explore the complexities involved when large Danish organizations bring together IT developers across cultures and geography to engage in collaborative work within software outsourcing setups.

Global software development research in Computer-Supported Cooperative Work (CSCW) and related areas typically distinguishes between open-source software development (see for example Daniel et al., 2013; Fugelli et al., 2013; Marlow and Dabbish, 2013) and corporate software development (see for example Boden et al., 2007; Espinosa et al., 2007; Herbsleb, 2007; Avram et al., 2009; Boden et al., 2009b; Jensen and Bjørn, 2012; Prikladnicki et al., 2013; Søderberg et al., 2013; Esbensen and Bjørn, 2014; Bjørn et al., 2014a). Open-source GSD is "an example of a peer production community fueled by volunteer contributors interacting, via computer-mediated channels, from all over the world" (Marlow et al., 2013, page 119). By contrast, corporate GSD projects, which are the focus here, are not voluntarily driven but merely rest on the companies' incentives of reducing cost and improving competitiveness. Taking a Danish perspective on GSD, this approach may be well-combined with the troubling and present issues that IT companies face today, namely to acquire enough domestic resources with the right set of technical skills.

In 2017, *Dagbladet Børsen*—a well-regarded newspaper that has existed for more than 100 years and that specializes in delivering business news in Denmark—published an article with the title: "Screaming shortage of IT people: TDC [the largest telecommunication company in Denmark], KMD [one of the largest IT and

software companies in Denmark] and others look towards East."[1] As consequence of the increasing demand for workforces with IT skills, universities have over the last eight years expanded their intake of IT students by 26%.[2] Companies compete side-by-side in attractiveness in order to recruit new young talents, often even before these students have left the universities. In the meantime, companies see no other options than to establish elements of outsourcing to avoid losing potential tender competitions and public procurement contracts. KOMBIT is one of the public tender providers that—on behalf of local government authorities—orders and procures IT solutions and handles the tender competition processes in Denmark. When publicly announcing the winner of one of their tender contracts, KOMBIT specifically highlights how the winner was evaluated as offering: *"The most economically advantageous tender."*[3] This resonates with the data that I collected for this dissertation; several managers in the Danish IT companies said something like: *"[…] It is a question of being able to compete on the hourly prices and that is why you need to use cheap labor from outside of Denmark—otherwise you will not win the contracts."* Based on statements like the one above and the grounds in which tender contracts winners are appointed, the discourse seems clear: if the Danish companies want to retain their market position and competitiveness, they need to utilize some form of outsourced labor workforce in their business model.

The cooperative arrangement of GSD in corporate settings is often referred to as 'outsourcing'. *Outsourcing* by definition refers to the contracting of work to an external partner, regardless of global economic issues, whereas *offshoring* means that companies seek to get work done in another country to leverage cost advantages. Global software development work in corporate settings is often realized by establishing an in-house offshore facility (a software development center), enabling companies to extend their own organizations by setting up subsidiaries or captive centers in countries with lower wages. Thus, when client companies subcontract activities to an overseas and independent third-party IT vendor, it is more precise to say that a company is engaged in *offshore outsourcing*

[1] "Skrigende mangel på it-folk: TDC, KMD og andre søger østpå" - link: http://borsen.dk/nyheder/avisen/artikel/11/163531/artikel.html#ixzz4c9dpKrZT.
[2] https://ufm.dk/uddannelse/statistik-og-analyser/sogning-og-optag-pa-videregaende-uddannelser/2018/notat-6-it-uddannelser.pdf

[3] "**Leverandørkontrakt tildelt på Kommunernes Ydelsessystem**" - https://www.kombit.dk/nyheder/leverand%C3%B8rkontrakt-tildelt-p%C3%A5-kommunernes-ydelsessystem

(Carmel and Argarwal, 2002; Vlaar et al., 2008), which is also the setup I investigate in this dissertation. The arrangement of GSD can take many forms depending on the incentives that drive forward the engagement (e.g., cost reductions or leverage of qualified labor resources). Moreover, the underlying organizational structures may involve, for example, a pricing model for how and at what price the GSD work should be undertaken. For instance, *time and material* is a pricing model whereby hours are ordered with a certain limit of team members associated, whereas a *managed service* model entails handing over a complete system or area of work for the supplier to maintain and develop. I have primarily investigated collaborative work organized on the *time and material* model through offshore outsourcing arrangements within Danish IT companies and their global IT vendors.

This research is based on three independent fieldwork studies conducted in three Danish IT companies, which operate within different business domains. All three companies are among the largest IT companies in Denmark and all engage in global software outsourcing to offshore locations in India and, to a lesser extent, in Poland, which makes them representative examples of GSD as it currently occurs within the Danish IT industry. While some of the globally distributed teams in these companies have managed to establish high-performing and productive collaborative relationships, there are still GSD projects that fail to achieve successful collaboration, which is also evident in related CSCW research on GSD (Boden et al., 2009b; Matthiesen et al., 2014; Matthiesen and Bjørn, 2017).

Throughout my doctoral research, I have taken an empirical, ethnographic approach to get a deep understanding of how politics and workplace realities converge in GSD. This perspective was not, initially, the one I intended to adopt. However, as I delved into exploring the nature of developing software in GSD, I found myself in a position where I was constantly reminded about the various political, cultural, and economic tensions that live alongside the cooperative work arrangement in GSD (Prikladnicki et al., 2013). In particular, when trying to unpack collaborative challenges through empirical narratives in relation to core CSCW concepts such as articulation work (Rönkkö et al., 2005; Boden et al., 2014), coordination (Cataldo et al., 2006; Boden et al., 2007; Herbsleb, 2007; Avram et al., 2009; de Souza and Redmiles, 2009), common ground (Olson and Olson, 2000; Bradner and Mark, 2002),

and closely coupled work (Bjørn et al., 2014b; Jensen and Nardi, 2014) the underpinnings of the collaborative challenges kept being referred to as issues of cultural differences by the IT professionals.

A main theme that surfaced in the data of the investigations of GSD presented in this dissertation is the challenge for IT developers to articulate their experience of the day-to-day practices of software development as they get entangled with cultural differences. The performance of cultural differences is subtle and yet powerful in how it shapes collaboration in GSD, as illustrated below:

> *"There is no doubt that the Polish consultants work very independently and challenge everything you say... it is not always the way the Indians do it. [The Indians] they just say "yes" and then they go on and develop what they think you want and then it is not always what you expected—instead of asking and such [...]"* (25.07.2013, Senior Software Developer, Interview, Denmark)

Complex interactions are at play, and they require careful detangling to understand how transnational encounters of GSD in Poland, India, and Denmark produce distributed collaboration as shaped around cultural difference. While the quote above may be read as more hostile than it was actually pronounced by this senior software developer who had several years of experience working with global outsourcing and particularly with consultants from India, it nevertheless captures a trend in GSD, which commonly accepts the use of controversial statements that include assumptions about how remote collaborators' national origins and cultural practices are linked to certain competences or work efforts.

In my empirical studies, the Danish collaborators describe their experiences as more challenging when collaborating with IT professionals in India than with IT professionals from Poland, and the variety in preferences and ways of working and collaborating are largely traced back to questions of culture. If we look for reasons for this trend, one approach that several companies apply when engaging in global collaboration and transnational work is to introduce various cultural training courses for their staff (Krishna et al., 2004). In the studies investigated here, such approaches had also been introduced, to some degree, through introductory cultural courses, keynote talks, or web resources that describe the cultural behaviors of and differences between, for instance, Indians and Danes. Moreover, in the three

empirical studies, the general consensus was that the staff in Denmark find that their managers have failed to equip them with sufficient tools for engaging in globally distributed collaboration (Matthiesen et al., 2014).

The problem with the kind of cultural training programs that are often presented in GSD is that the explanations of different cultures tend to point out highly static and general descriptions of peoples' behavior attributed to their nationality. In particular, fellow scholars criticize the foundation for such training as it is based on the essentialist idea that collective behavioral patterns can be ascribed to an entire population (Søderberg and Holden, 2002; Walsham, 2002; Kwek, 2003). From a research perspective, describing collaboration issues through general and stereotypical descriptions of cultural differences is also problematic when trying to analyze and understand collaborative practices such as knowledge-management practices (Boden et al., 2009a), as well as to design future tools for collaborative support, which is core to CSCW. In CSCW, ethnography has contributed by gaining insights into understanding the 'nature of work' and the actual work practices that take place when people collaborate using computer technologies. Therefore, when ethnographically exploring the domain in its 'natural settings' and using the vocabulary from the field (Blomberg and Karasti, 2013), it is important to think about how to analyze and probe beneath the surface of the collaboration-related challenges that are interpreted through negative stereotypes of national culture.

In this dissertation I offer an alternative lens and vocabulary for illuminating and discussing collaborative issues that risk being reduced to negative stereotypes that fit normative descriptions of national cultural differences in GSD. I develop a GSD Stereotype Framework to analytically address negative stereotyping as it takes place in GSD and develop insights on the specifications that underly the concrete coordination and communication challenges in developing software collaboratively. The GSD Stereotype Framework describes the key areas and attributes that are important to attend to in order to understand the real-life and situated collaborative challenges that those involved in GSD work experience daily. The question of the dissertation revolves around the mundane ways that negative cultural stereotypes enter GSD practice.

2 Research Question

How can researchers and IT professionals move beyond negative stereotyping and instead address the concrete coordination and communication practices that cause problems in global software development?

3 GSD as a Collaborative Endeavor

One of the fundamental drivers for engaging in GSD is to recruit and engage skilled labor from lower-income countries. However, the expected saving of development costs has been questioned by scholars who argue that cost should not be the primary focus, but instead offshoring can be justified for other reasons such as flexibility or freeing up local staff to expand the development portfolio (Šmite and van Solingen, 2016; Šmite et al., 2017). Nevertheless, several political tensions come into play for GSD. One tension revolves around the *resistance* that companies may encounter from their existing staff who feel insecure about their future jobs and careers (Rost, 2004), as the existing staff must find ways to improve their skills and qualifications in order to maintain ownership of their work, code, or market value (Metiu, 2006). While resistance against global outsourcing may arise among those people who need to change work practices from local to engaging in globally distributed software work, this resistance cannot solely be attributed to the people 'on the floor' or serve as an argument for why things go wrong (Matthiesen and Bjørn, 2017). Instead, resistance may arise upon the management's lack of attention to the way outsourcing is initially introduced and supported afterwards (Matthiesen et al., 2014).

Distributed work in GSD is inherently a collaborative endeavor (Prikladnicki et al., 2013) and, due to many IT systems being highly complex and interlinked in practice (Bjørn et al., 2014a), software development requires a range of different IT professionals to work together to deliver a software system with the desired specifications within a certain budget and time frame. For the same reasons, those involved in GSD work often face core challenges in handling, sharing, and making sense of the required knowledge for developing a complex system. Thus, software development work depends on several tools and artifacts such as requirement specification, test cases, integrated development environments (IDE), design diagrams, product backlogs, project management plans, database tables, and UI designs among others. Developing software systems within temporal and spatial boundaries is non-trivial and, consequently, over the past two decades GSD has become a research theme of interest in various disciplines. The focus on global aspects within software engineering and information systems has turned into a GSD

research sub-field with its own annual conference, focusing largely on methods for assuring quality in the product by improving and streamlining software processes and tools (Avram et al., 2009). While the literature on methods for standardizing and defining more effective software models, tools, and engineering processes offers important insights for making GSD work more efficient, many studies offer little insight into the actual conduct of work and work practices undertaken by human actors involved in GSD. Therefore, the area I expand on here is within the interdisciplinary research field of CSCW, which seeks to analyze and understand the basic nature of work with the aim of designing collaborative technologies (Schmidt and Bannon, 1992). In particular, I seek to understand the distributed collaborative work of GSD that takes place within IT Denmark-based companies that are engaged in outsourcing IT development work to IT vendors located in India and Poland. To this end, it is important to explore the actual needs and requirements (in addition to formal descriptions of work) that people have to deal with when they engage in distributed collaborative work.

While the various research fields interested in GSD have different key interests and methods (Avram et al., 2009), it is commonly accepted that GSD work or distributed work in general is challenging due to the temporal and spatial distribution of work (Bjørn et al., 2014b). Ever since Olson and Olson (Olson and Olson, 2000) published a paper on how it is difficult to work across distance, this has been common point of departure in literature on distributed work (Bjørn et al., 2014b). Nonetheless, scholars have proposed that it is not the physical distance that complicates collaboration but instead the *perceived distance* (Bradner and Mark, 2002), as it has been found that people tend to act and present themselves differently depending on how far they believe they are located from each other. Bradner and Mark found that people: 1) tend to use extreme descriptions (being deceptive) with people they believe are far away; 2) are more easily persuaded by people who they believe are closer geographically; and 3) are (initially) less likely to cooperate with people they believe are geographically far away. This finding is interesting as it suggests that aspects of more situated, individual, and socio-emotional characters should be considered when trying to understand the nature of the work and the real-life practices that people involved in GSD engage in on an everyday basis.

Certain boundaries characterize distributed collaborative work such as *time*, *space*, *organization*, *profession*, *language*, *culture*, etc. However, Watson-Manheim et al. (2012) have pointed out how these boundaries are considered uniformly problematic even though certain boundaries are not always perceived as discontinuities. They suggest that a *discontinuity* "is created at a boundary when an individual perceives a change in information and communication flows that requires conscious effort and attention to handle" (Watson-Manheim et al., 2012, page 36) and it is only when those people working in distributed settings perceive a boundary to be a discontinuity that this boundary become problematic. This means that, in order to understand and analyze challenges within GSD work and work practices, it is not sufficient to consider distance as a physical boundary that people either manage to cross or not; instead, distance is one of the many discontinuities that present a challenge to working in distributed settings. Therefore, it is vital to assess the dynamic ways in which time, space, culture, but also software practices, knowledge, domain language, and professional expertise (Matthiesen and Bjørn, 2017) emerge at the various boundaries in GSD. Through this analytical perspective on discontinuities at the boundaries in distributed collaborative work, a simplistic and dichotomous lens can be substituted for more dynamic one. This dynamic perspective can help analyze and understand the underlying issues and potential problems in GSD, as well as the concurrent continuities that can help in providing common ground and cohesion to those participating in distributed collaboration (Watson-Manheim et al., 2012).

3.1 Articulation Work in Distributed Work

In CSCW, collaboration is when multiple individual collaborators are mutually dependent in their work (Schmidt and Bannon, 1992). When analyzing collaborative work, there is one important distinction that is useful in understanding the multiple and messy activities that collaborators need to engage in in order to get work done. This is the distinction between 'work' and 'articulation work', which is seamlessly integrated in a collaborative practice (Schmidt and Bannon, 1992). *Articulation work* is defined as all the additional work required for performing the work. For instance, when collaborators need to coordinate, divide, allocate, and schedule their distributed yet individual activities (Gerson and Star, 1986; Strauss, 1988b), this is considered articulation work. However, the boundary between what counts as work

or articulation work is not clear-cut; instead, this distinction depends on the professional identity of the individual collaborator. A project manager may, for instance, consider the work of planning and monitoring a software project to be the 'work', while the same activities may be considered as the 'articulation work' for a software developer.

Geographically distributed working can affect the nature of colleagues' interactions, as communication is primarily computer-mediated and conversations may be conducted in languages that are not necessarily native to the participants. Thus, to enhance the chances for succeeding with collaboration across distance, Olson and Olson (2000) emphasize the importance of common ground. C*ommon ground* refers to the knowledge that two or more people have in common—and this is knowledge that they *know* they have in common. This concept is related to the collaborative practice of 'grounding in communication', which is a process that Clark and Brennan (1991) identified when investigating how people establish and negotiate mutual knowledge, beliefs, and assumptions in a conversation. The *grounding* process is essential to communication, as it is essential that the messages we formulate and exchange with one another are understood as we intend them to be. Depending on whom we are talking to, we make some assumptions about what this person knows and, based on these assumptions, we frame what to say.

The challenges of establishing common ground and handling the effort of articulation work in GSD are complicated. Global software development work often involves hundreds of people with different professions, speaking different languages, and collaborating across time and geography. Moreover, developing software is a process that entails significant interdependence in terms of tasks (Herbsleb et al., 2000; de Souza and Redmiles, 2008) as well as expertise (Faraj and Sproull, 2000; Matthiesen and Bjørn, 2015). Thus, when handling, sharing, and making sense of the required knowledge for developing a system, the work includes having a common language and terminology for the different methods, tools, and artifacts used in a project, and following common communication processes. However, this is not easily accomplished, and the allocation of tasks and the distribution of professional roles matters greatly for the distributed collaborators' ability to establish common ground (Matthiesen and Bjørn, 2017). Moreover, the

tools for mediating cross-site communicating in GSD may be put into practice in different ways and, as a consequence, legitimate efforts of performing articulation work and establish common ground in GSD risk being perceived as interruptions (Matthiesen et al., 2014).

To further reduce the efforts of articulation work and enhance the chances of succeeding with collaboration across distance, Olson and Olson (2000) have also highlighted *coupling of work*, which refers to the level of ambiguity there is in the collaborative tasks. The more ambiguous a task is, the more frequent and complicated interaction is required, and the more tightly coupled the work at hand becomes. To succeed in distributed collaboration, Olson and Olson (2000) argue that chances are improved if tasks are loosely coupled by nature or coordinated in such ways that the coupling becomes loose. Obtaining loose coupling between tasks fits well with one of the most influential principles in software engineering, which concerns decomposing a software system into smaller pieces (components or modules) in order to manage dependencies in the system and thus between software tasks (Parnas, 1972). Through modular decomposition, the proposition is that people are able to work more independently and with fewer communication and coordination needs. While there are benefits in enabling people to work in parallel, modular decomposition has recently been questioned in reference to collaboration as it is argued that this principle does not necessarily reduce the developers' need for coordination (de Souza and Redmiles, 2009). Instead, instantiations of this principle may hinder some forms of collaboration as teams lack awareness about the dependencies that cut across modular components (de Souza et al., 2004). In addition, loose coupling among software components does not only offer collaborative challenges during the development phase. Grinter highlights the challenges of managing dependencies among components during *recomposition,* which is the coordinative work of reconnecting and reassembling components into a whole software system (Grinter, 2003). This is particularly challenging as the work necessary for recomposition is manifested in the communication and responsibilities of individuals within the organization.

The otherwise commonly accepted benefits of modular decomposition come with a set of disadvantages. The fundamental question in this respect concerns whether (and what kind of) dependencies are identified, as this will determine the nature of

the necessary coordinated (Cataldo et al., 2006). However, even if dependencies are identified at the design phase, dependencies change and evolve throughout the development of the IT system. Within small-scale and co-located software projects, the identification of work dependencies may be less vital as people typically do articulation work and establish common ground in multiple situations without even realizing it. Articulation work may take more informal or coincidental forms; for example, during lunch or by visiting a collaborator's work desk. Articulation work in GSD, however, requires more formal arrangements and taking time to do articulation work is not easy to arrange. However, it appears that closely coupled work tasks can encourage collaborators to commit to the distributed interaction and collaboration (Jensen, 2014; Bjørn et al., 2014b). In particular, when collaborators from various sites are highly dependent on each other's work, they are more willing to perform articulation work and establish common ground.

3.2 Coordination in GSD

As described above, one way of handling articulation work is through collaborators' everyday social interaction. Yet, when the number and the distribution of the involved collaborators increase, collaborators lose some of the rich and subtle interaction that co-located teams use when managing dependencies among tasks (Herbsleb et al., 2000). Therefore, handling the complexity of articulation work in distributed settings requires more structured mechanisms of interaction such as organizational structures, workflow systems, schedules, etc. One way to reduce the complexity of articulation work is through *coordination*, which can be achieved by scheduling and planning in ways that make things work together (Gerson, 2008). The strategies for coordination and ways to handle articulation work include trying to do more with the same resource or do the same work with fewer resources. This may be achieved by, for example, breaking complex tasks into multiple independent tasks with the goal of eliminating some relationships within a complex task. Another strategy is by keeping things the same, based on, for example, a classification scheme that helps make relationships uniform. Nevertheless, articulation work still needs to be done as unanticipated cases occur, hardware fails, bugs emerge in software, people go on vacations, mistakes are made, etc. (Gerson, 2008).

Managing dependencies among software tasks is one of the most challenging activities due to the dynamic nature of software development (Cataldo et al., 2006). In addition, GSD work is recognized as software development with additional complexities in terms of handling the extra effort of articulation work (Avram et al., 2009) and, thus, coordination is considered a crucial enabler for GSD. This means that there is great interest in understanding the use of and developing technologies that can handle issues related to managing the dependencies in GSD (Herbsleb et al., 2000; de Souza and Redmiles, 2008). Coordinative technologies include a range of general-purpose communication tools such as instant messaging, group chats, social media, or email (Boden et al., 2007). Regarding open-source technologies specifically, there are social code repositories such as Github (Dabbish et al., 2012), which enable highly skilled professionals to work together and follow certain structures when developing a large-scale system. Within corporations there are also more domain-specific tools such as process and planning tools, IDEs, integrated task tracking systems, bug tracking systems (Avram et al., 2009), burn down charts (Matthiesen and Bjørn, 2017), and visualization tools (Halverson et al., 2006). Many of these tools may, to a certain degree, help with managing dependencies in software work through coordination. However, ethnographically informed studies of the use of coordinative tools to support articulation work have identified how "actual work practices sometimes differ from the perceptions that managers have about software development strategies in their companies" (Boden et al., 2007) p.243, and these differences sometimes have dire ramifications for the way work is coordinated and monitored in a project (Matthiesen and Bjørn, 2017). These discrepancies between software development strategies and how software work is accomplished in practice lead into the next section, which explores what underlies the technical solutions for supporting the collaboration that is a key to CSCW.

3.3 Categories and What They Do for Us

Categories play an important role when designers, engineers, or software developers try to capture and support collaborative mechanisms in technological tools and artifacts. Whenever a technological system is designed and built, one of the first things that is created is a model that serves to inform and help identify important relationships, requirements, dependencies, etc. in the domain wherein the system should operate. When things are put into a set of classes or categories, we

have a certain classification scheme that can do some kind of work for us (Bowker and Star, 2000). A classification scheme can help us arrange or sort things into 'kinds of things' according to common relations, affinities, similarities, or differences, and this can be useful when trying to comprehend a new domain. Oftentimes the categories in a system are invisible in use, but become visible when the system breaks (Bowker and Star, 2000) or fails to serve its purpose.

A common pitfall for how technical solutions have been thought of to support collaborative work in various domains is linear thinking, whereby formal descriptions of the work (Suchman, 1995; Blomberg et al., 2003), business processes, and workflows are translated one-to-one without taking into account the actual "flow of work" performed within a practice such as GSD (Avram et al., 2009). For instance, to support the coordination needs emerging among collaborators when shifting from co-located software teams to distributed teams, such translation may include that a physical collaborative artifact such as a paper or spreadsheet is turned into a digitized version. These translations do not always work well, despite the fact that researchers have tried to develop sophisticated representations of whiteboards, for example, or low-level spreadsheets to serve as digital scrum board prototypes (Esbensen et al., 2015). The reason for this ongoing failure of technical solutions may be because the new artifacts fail to take into account the concrete work practices as well as the physical surroundings in which a new technology becomes situated. An example of this is when, in the 1980s, Winograd and Flores presented The Coordinator, a collaborative technology to support everyday communication within the workplace (Winograd and Flores, 1987). Their system was an early attempt to explicate basic contextual information around the communication between collaborators in a work situation by enabling collaborators to annotate the meaning of their communication through certain categories.

The categories upon which The Coordinator was built emerged from traditional cognitive theory and this approach gave rise to a range of interesting and ongoing discussions on what are or should be the foundations of computer systems for human interaction. In particular, anthropologist Lucy Suchman worried about the origin of the categories implemented in The Coordinator, as these did not emerge based on how people communicate within a practice. This concern set in motion a

discussion on what categories can do for us, and how they can be political and enforce mechanisms of control and social order. On that note, Suchman was inspired by Winner who had earlier critiqued the potential political, racial, and social biases that were built into the architectural design of the low-hanging bridges over the parkways in New York (Winner, 1980). The height of the bridges enabled cars—primarily owned by the white upper class—to reach recreational areas on Long Island, while public busses—primarily occupied by poor or black people—could not use these bridge underpasses. While Winner's argument was about how technological artifacts build order in our world and order human activity, this intentional or unintentional disciplining or favoring of human activity is equally relevant when investigating the way globally distributed collaboration is supported through technological tools or how efforts of work are represented in collaborative artifacts (Matthiesen et al., 2014).

Since categories build upon conscious and unconscious assumptions about the world, society, culture, and domain wherein they operate, categories are never neutral (Bowker and Star, 2000). As with technological artifacts, categories and classification systems have agency to invoke political agendas, since they have the potential to enable, exclude, marginalize, render invisible, or constrain people, animals, environments, or systems (Suchman, 1993; Orlikowski, 1994; Sengers et al., 2005). In this way, categories can be used consciously and unconsciously (Sengers et al., 2005), and, thus, in the utilization of classification schemes an important question is: what are the basic assumptions from where categories emerge and become implemented in the design of cooperative systems? In the discussions on technologies for highly technically skilled professions (such as GSD), it is therefore important to investigate how certain categories serve as enablers or constrainers of human activity in the workplace. In particular, when classification schemes for collaborative systems fail to represent the messiness of our world, this can have consequences for the people working with the systems. This is important for the people who handle the implicit articulation work (Strauss, 1988a), the normal natural trouble (Randall et al., 2010), or the informal work that is not part of their formal job description (Star and Strauss, 1999; Star and Bowker, 2007; Blomberg and Karasti, 2013); these people (and their work) risk being overlooked, unacknowledged, or even rendered invisible.

Categories that are not formally represented within a classification system have been coined as *residual categories*. These are characterized by broadness, whereby details, contingencies, and variations risk being silenced, which may affect the everyday lives of people inhabiting a residual category (Star and Bowker, 2007). A core value for the design of cooperative systems should then entail responses to and support of formal descriptions of work, while also taking into account the implicit, unspoken, and important work of an informal character that makes up the residual categories in GSD work. However, in understanding GSD work in relation to people, practices, and technologies, we are forced to further our understanding of the tensions and, thus, the categories that underlie arrangements of GSD. In particular, the categories of analysis offered by Western theories' interpretation of cultures have been criticized for being blind to their own ethnocentrism, and thus there is a need for giving voice to alternatives stories that can shed light on, and critically re-evaluate, Western theories' dominant discourses of cultural and social marginality (Kwek, 2003). Nevertheless, when these categories are too limited, there is a risk that important aspects of global work or workers can be overlooked or marginalized through the adoption of collaborative practices and system designs that reinstate inadequate assumptions or stereotypes about the global work. Moreover, when responsibilities and development tasks are distributed between collaborators, the highly practical and potentially power-laden considerations involved (Metiu, 2006) may be based on assessments of whether tasks are vital (important), innovative, or technically trivial (Vora, 2015). Thus, in the efforts to address issues in GSD work it is crucial that we investigate the existing and residual categories within the particular collaborative arrangements in GSD and how the people, technologies, and practices perform these categories.

To meet the challenges outlined above, it is important to apply multiple gazes that can account for the many different perspectives of those who take part in the global work. These angles include an agenda of moving away from Western ethnocentric or normative interpretations of cross-cultural collaboration, as well as colonial legacies of power and labor allocations between the 'Global North' and 'Global South'. Depending on whose gaze we adopt, different aspects of GSD work and technologies will come to our attention as we determine the most important or

proper practices. Here, issues of power and privilege come into play, and these are closely tied to formal roles and the ways in which structures and practices are enacted in the collaborative work (Hinds and McGrath, 2006). For example, the physical location of one's body relative to where the work is performed (D'Mello and Sahay, 2007) or its hierarchical position (Hinds et al., 2015) in the collaborative work matter for how one perceives the role of cooperative technologies and how these should support the people and practices within GSD. In addition, the experience of entering global collaborative work is significantly different for those workers who are hired based on the global premise in comparison with those who have been working in co-located IT projects for decades. The perspectives on what GSD work can bring about in the future varies; to some it is a career opportunity while others fear that they are slowly being pushed out of the labor market. Thus, investigations regarding from where and by whom the global work is performed should be undertaken, with multiple angles that account for what global work entails and for whom. Such complex investigations are important for creating better accounts of the globalized, culturally heterogeneous, and locally specific (Walsham, 2008) world that GSD workers design for.

The contemporary conditions that affect globally distributed work, including the use and design of technological systems in GSD, have been further elaborated on within a stream of human-computer interaction (HCI) research coined as transnational HCI (see for example Shklovski et al., 2010; Castro and González, 2014; Shklovski et al., 2014; Bjørn et al., 2017). Transnational HCI is concerned with understanding the ways in which "local experience and remote connectedness interact in ways that are sometimes complimentary and sometimes a source of tensions and problems as people construct, navigate and manage boundaries between them" (Shklovski et al., 2014, page 15). This is relevant to the attempt to understand GSD work from multiple angles to uncover concrete issues and address negative stereotypes. When it comes to the categorization of *culture*, the transnational approach suggests sensitivity toward the word 'culture', building on the perspectives of the social theorist Appadurai. To him, culture—when considered as a noun—has certain implications. Culture becomes this thing or object that people have; a property of people or places; a substance equated to ethnicity, which seems to "bring culture back into the discursive space of race" (Appadurai, 1996, page 12). Appadurai argues

that culture in adjectival use is more useful as we can say that a practice, an object, or an ideology has a cultural dimension, which attends to the local, embodied, and situated differences (Appadurai, 1996). Therefore, when investigating collaborative challenges in GSD and considering how these are linked back to culture, we should take into account that it matters greatly from where and by whom GSD work and practices are considered.

4 GSD from a Cultural Perspective

In combatting cross-cultural issues within transnational collaboration, the most applicable, and thus pervasive, strategies we have seen within the field of GSD as well as in classical cross-cultural management literature relate to the *cultural dimensions*. The foundational research here was initially developed by Hofstede and colleagues and was based on survey data collected in IBM back in the 1960s and 1970s (Hofstede, 1984; 2010). In particular, the variety in preferences and ways of working and collaborating are often superfluously explained through the adaptation of Hofstede's cultural dimensions, which map out people's behavioral patterns and work attitudes based on their ethnicities and national origins. A few fellow scholars have offered contributions that take on an alternative gaze on culture by focusing on the ways in which national, professional, and organizational aspects of culture are manifested in artifacts and practices of GSD (Boden et al., 2009a). Moreover, prior research has highlighted that using the cultural dimensions to explain the challenges in GSD produces problematic side effects. In particular, when participants are allowed to use 'culture' as a rhetorical strategy, shying away from reflectively considering the emerging problems, the complexity involved in developing software within GSD team collaboration becomes hidden and reduced to cultural-deterministic language (Jensen and Nardi, 2014).

As a strategy to combat national culture differences, organizations often introduce cultural training programs based upon Hofstede's cultural dimensions of the countries involved in globally distributed collaboration. Such training events introduce the general descriptions of peoples' collective behaviors based upon their nationalities. In GSD, the cultural dimensions most commonly applied include hierarchical power structures and inequality (power distance), dealing with

uncertainty, and the extent of people's preferences for high levels of predictability in their work (uncertainty avoidance), as well as whether people have strong ties to people belonging to the same community (and thus, act in a collectivistic or individualistic way). Nevertheless, such strategies have been criticized by various researchers (Søderberg and Holden, 2002; Walsham, 2002; Kwek, 2003), since the foundation for such training builds on the essentialist idea that an entire population can be attributed collective behavioral patterns. Indeed, these particular patterns can be constantly reinstated and confirmed when they are applied as explanatory facts.
One might wonder why the application of Hofstede's cultural dimensions is still dominant when organizations try to comprehend and resolve their cross-cultural challenges, despite the many critiques. The reason why these dimensions have become part of the acceptable management vocabulary (Kwek, 2003) is not only due to a lack of alternative approaches or theoretical framing in the effort to challenge the underlying assumptions and practices about work across national boundaries. Instead, a key reason for the popularity of the dimensions is the understandability of the framework and the ways in which generic dimensions are easily applied, explained, predicted, and confirmed (Kwek, 2003). When behavioral patterns and collaboration issues are explained in terms of, for example, power distance or uncertainty avoidance, and then these explanations seem to be confirmed by Hofstede's framework, researchers are then tempted to continuously revisit and reapply the same explanations and categories.

Such simplistic descriptions of culture maintain the status quo by re-inscribing stereotypes that were developed more than 30 years ago. The world has changed in the last decades; technology has enabled people to engage in closely coupled collaboration despite being separated by geography and/or time. While generalizations may mitigate overall misunderstandings in the collaborative work, they offer little help with trying to govern and orchestrate complex collaborative work settings such as those we encounter in GSD. The ways in which considerations of cultural differences have been applied in GSD so far still remain slippery as we may in fact encounter difficulties when trying to categorize culture as a substance and thus something we 'have' (Appadurai, 1996). In particular, the question then becomes: how do we categorize the 'culture' of a senior developer who was born in India but earned her engineering degree in the US, worked for 10

years in Finland, and has now returned to India to care for her parents while working at a subsidiary of an international company based in Denmark? In other words, when trying to understand the real-life and situated work practices—which is what I am interested in here—culture seems only minimally helpful for understanding the concrete and real-life challenges. Instead, the stereotypes that we reinstate about our foreign collaborators' behaviors run the risk of becoming self-fulfilling behavioral patterns through what psychologists describe as *confirmation bias*.

Confirmation bias refers to the ways in which we make decisions in a way that is largely designed to confirm the beliefs that we already have (Nickerson, 1998). This means that we seek or interpret situations of evidence that support our existing belief systems, expectations, or hypotheses. So, when we encounter statements such as: "In India it is not common for people to question an assignment, even if they don't understand it," it is vital that we do not take such statements at face value and explain all collaborative situations based upon generalized statements by a developer from India. More importantly, we need to dig deeper into the experiences we encounter but find odd, if we are to understand the reasons behind the lack of common approaches across geography toward completing a programming assignment. We should ask questions such as: is there a fundamental gap in the comprehension of the work at hand or in the ways in which we distribute and handover task assignments? Or could the stereotypical description of the '*Indian behavior*' be related to the ways in which power or agency is distributed in the collaboration?

4.1 Implicit Bias

The existence of cultural differences need not be dismissed entirely. The work and work practices performed in GSD, for example, may entail cultural practices that differ within a globally distributed work arrangement. However, one of the risks of developing cultural stereotypes is that these accounts become too simplistic and, thus, counterproductive. Scholars have previously highlighted the need to develop a cultural intelligence that prevents us from devising simplistic explanations about the world but instead helps us make simplifications that we can use to navigate complex problems in the world (Ang, 2011). The dilemma in interpreting and navigating the

cultural differences in GSD lies in the space between, on the one hand, the ***simplistic*** explanations of national cultural behavioral patterns produced by cultural stereotypes and, on the other hand, the many desirable ***simplifications*** that can serve as an intelligent tool or strategy for working with and benefitting from the cultural differences we encounter in our everyday work with GSD. To address this gap, I explore ways to develop and adapt alternative strategies for understanding and combatting the challenges involved in working across various discontinuities in GSD.

In doing so, I introduce the concept of ***implicit bias*** into CSCW research on GSD. Implicit bias (or unconscious bias as it is also referred to) emerges from the field of social psychology. In the context of GSD, deploying considerations of implicit bias serves as a useful strategy for understanding the motives and mechanisms that ***implicitly*** influence the ways in which we organize and act in the world. Biases do not only apply to "*bad*" people who consciously demonstrate discriminatory behavior toward others.[4] Instead, biases apply to all of us; they affect the ways we act in the world. The traditional argument within social psychology is that people's actions are based upon their implicit biases, which emerge from the way people are mentally 'programmed' or hardwired to filter and make decisions quickly in order to survive. This argument builds upon research on sensory systems in neurophysiology, which posits that the human brain receives 11 million bits of information per second; however, it cannot process more than 40 bits per second (Zimmermann, 1986). As a result, when we see or perceive that something or someone is dangerous, the human brain deploys, on a subconscious level, a danger detector (LeDoux, 1996) that is activated even before we start thinking. While our ability to unconsciously filter and navigate among millions of inputs serves an important purpose, the same mechanisms can negatively influence the ways in which we interact with our colleagues in the workplace. We organize our social worlds through categorization, which is important and necessary for interacting with the world. We see patterns that are based on the accumulated effect of everything we have been taught or exposed to throughout our lives. However, these patterns are sometimes flawed and can contribute to comforting presumptions that

[4] http://www.cookross.com/docs/UnconsciousBias.pdf

re-inscribe stereotypes and confirm our expectations and biases (Fiske, 2000), causing us to neglect the complexity involved.

In an attempt to better understand our own (implicit) unintended preferences, psychology researchers from Harvard, Virginia, and Washington introduced in 1998 the implicit association test (IAT)[5] (Greenwald et al., 1998) with the purpose of detecting people's implicit associations (biases) toward, for example, race, gender, sexual orientation, national origin, disabilities, age, etc. Implicit bias has been investigated within various domains and research has shown, for instance, how healthcare providers' unconscious biases can influence their behaviors and judgments (Stone and Moskowitz, 2011), or how racial biases can influence medical decision-making (Paradies et al., 2013). Likewise, there are studies that show how various types of racial or gender discrimination take place in hiring situations. One study demonstrated how resumes with European-American-sounding names received 50% more interview call-backs than identical resumes with Latin-American-sounding names (Bertrand and Mullainathan, 2004). Another study demonstrated how merit was redefined to justify gender discrimination in a hiring and recruitment process; the criteria for success at the job was found to be implicitly redefined in order to fit the candidate with the desired gender (Uhlmann and Cohen, 2005). Finally, implicit bias has also been found to heavily influence the decisions that are made when recruiting or selecting applicants within research. Moss-Racusin et al. (2012) found that science faculties in higher education institutions, when reviewing identical applications with differing names and genders, had biases favoring male applicants over female. For example, male candidates were rated as better qualified than female candidates, and thus the science faculty was more likely to give the male candidate a higher starting salary and invest more in the development of the male candidate than the female candidate.

In recent years, major companies such as Microsoft[6], Google[7], and Facebook[8] have addressed the issue of implicit bias in an attempt to improve their recruitment and hiring processes, as well as general interaction in the workplace. These companies

[5] https://implicit.harvard.edu/implicit/takeatest.html
[6] https://www.microsoft.com/en-us/diversity/training
[7] https://rework.withgoogle.com/subjects/unbiasing/
[8] https://managingbias.fb.com/

are all highly dependent on success in welcoming and embracing high levels of diversity among staff as well as assuring cultural diversity and gender inclusiveness in the products they develop. A focus on implicit bias enables us to broaden our understanding and recognition of the blind spots and implicit biases that each of us carries around. Building on previous insights related to considerations of implicit bias, this dissertation develop new strategies for creating an understandable, useful, and comprehensible framework for addressing the challenges involved within collaboration in GSD.

5 Method

Given that the main research interest in this dissertation is to further our understanding of how to move beyond negative stereotyping and address concrete collaborative practices that cause problems in GSD, it was important to apply research methods that were appropriate for engaging empirically with various local contexts and settings of GSD to investigate the different coordination and communication complexities present. My methodological choices were empirically driven from the very start. To produce a deep understanding of the different collaborative complexities involved in developing software in global outsourcing, it was vital to explore the ways in which the work in GSD is performed in practice within different corporations and their involved locales. In my capacity as a contributor to the NeXGSD research project—of which this doctoral research is a part—I was able to gain access to and conduct empirical studies in three large Danish IT companies, which I introduce further in Section 5.1. To investigate GSD work practices and gain as much detailed insight into the various ways work and collaboration take place in real-life practices (Blomberg et al., 1993), I spent a lot of time at different field sites in Denmark, India, and Poland conducting several workplace studies (Luff et al., 2000) using ethnographic research methods (Randall et al., 2010). I found ethnography suitable for this research as the underlying assumption of ethnography is that, in order for researchers to gain an understanding of the world and make sense of it, they need to participate in social life and encounter the world firsthand (Blomberg and Karasti, 2013). Ethnography has been applied by anthropologists seeking to understand the design and evaluation of software since the 1970s, and the perception of ethnography as a useful tool in technology design was first established by the mid-1990s (Forsythe, 1999). Today, ethnography is widely used as a research approach in traditional and applied social sciences (Blomberg and Karasti, 2013).

Ethnography is the study of people in naturally occurring settings or 'fields' by means of methods 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, page 10)

For CSCW and other related fields, ethnography has played an important role in uncovering aspects of work that are otherwise invisible or undervalued (Suchman, 1995; Star and Bowker, 2007). Moreover, this approach has broadened our understanding of work as situated and socially embedded (Suchman, 2007). When conducting the research for this thesis, I kept in mind one of anthropology's key axioms: what people say they do and what they do are not always the same, as people may be unable to articulate—or unused to articulating—their activities in details (Blomberg et al., 1993). To gather insights into people's daily work practices and to understand *how* and *why* people do the things that they do (Sharp et al.), I spent many hours observing people in their daily work. This work included all kinds of activities, ranging from strictly planned tasks to more informal activities and interactions. I shadowed project participants for several hours. This involved, for instance, observing people while they worked at the computer, attended formal and ad hoc meetings (physical and virtual), walked around the office building to find a certain colleague, interacted via digital communication tools (instant messaging, email, or conference calls), ate lunch, or took a coffee break.

In an effort to convert detailed insight into concepts and premises that help us understand the things people do, I investigated the various ways people engaged in discussions with co-located colleagues as well as reflected upon or described global outsourcing to me or their distributed collaborators. I combined various data-gathering techniques—such as participant observation, document analysis, and interviews of both formal and informal character—with a careful analysis of the data and a continued awareness of my role as a researcher and ethnographer (Forsythe, 1999). I recorded this awareness of my role in fieldwork diaries and confessional statements that served as small testimonies of my personal experiences in the field and enabled me to expose and reflect upon my actions and failings (Schultze, 2000). When I was still in the field, this approach helped me keep track of the direction of the research as I constantly accounted for and analyzed the decisions I made underway. These decisions involved considerations of what themes I should delve further into; how I should try to position myself to build trust and encourage the informants to speak more freely about their work; or how my presence impacted field. After I left the field, these fieldwork accounts were equally useful when I revisited my empirical material for further analysis and when I was trying to

understand how I had developed as a researcher. I tried to focus on the way collaborators involved in GSD interpreted various collaborative activities, issues, or clashes, and to then combine people's interpretations and meanings with the dynamics present in the various empirical studies (Eisenhardt, 1989; Walsham, 1995). In this way, I set out to understand the complex whole by iterating between the interdependent meaning of smaller events in the empirical data and the whole that they formed—an approach inspired by hermeneutic principles (Klein and Myers, 1999). So, the research that goes into this dissertation is a product of a constantly evolving, iterative, and reflective process where I have shifted between conducting fieldwork, producing and revisiting existing field data material, and discussing preliminary findings or inconsistencies in the data with the empirical field as well as with research colleagues.

Due to my key interest in understanding the actual work and work practices in GSD, I went into the field without a rigid or pre-formulated research question at hand (Forsythe, 1999). Instead, I had developed an overall strategy for my ethnographic approaches, which involved several ideas, questions, and assumptions based upon existing theory and concepts within the domain. This 'ethnographic strategy' (Neyland, 2007) was far from a fixed plan, but it served as a helpful device for maintaining a coherent focus in executing my research and trying to handle interesting opportunities, directions, or problems that emerged. The concepts that I brought with me included articulation work, common ground, and closely coupled work, which are concepts of key interest for CSCW scholars and which involve exploring and understanding the ways in which distributed collaborators communicate and coordinate their interdependent yet individual work. I did this by strategically focusing on the multiple dependencies that exist among various systems, software tasks, and people involved in developing large and complex IT systems in distributed arrangements. I constantly calibrated between the theoretical concepts and the real-world social processes (Forsythe, 1999) that are in play when people collaborate across various discontinuities in global outsourcing. This means that I carefully investigated *when* people engaged in articulation work within both the local and the distributed settings and *what* tools they used for mediating their communication. Articulation work can take form in various ways: it can be informally or formally organized, and it might be perceived as more or less

important by those involved in performing the articulation work. This means that one person's articulation work may be perceived as a burden by another person due to the degree to which people are coupled in their work tasks as well as to what degree those involved have common ground. In this way, exploring articulation work as it takes place in actual work situations illuminates the aspects of collaborative work that relate to the way work is organized and distributed as well how certain power relationships come into play. Nevertheless, these aspects are less obvious or even invisible if considered through formal as well as informal descriptions of work, which is why empirical investigations in real-life settings are so important to understand GSD work from a broader perspective.

In my theoretical positioning I stressed the importance of re-evaluating the categories of analysis that underlie the way we understand practices in global work and design technologies for its support. In order to do such re-evaluations, challenging the current take on and the vocabulary used in GSD is appropriate, as it seems there is a need for alternative perspectives on what has previously been dominated by discourses of Western ethnocentric interpretations of culture. In stimulating new ways of addressing issues of power, inequalities, and privileges in globally distributed software work (see for example Hinds et al., 2015; Matthiesen and Bjørn, 2016; Metiu, 2006; Jensen and Nardi, 2014), we need to understand particular design contexts and critically reflect upon unintended consequences in the designs while simultaneously avoiding reproducing or reinforcing status quo of well-known stereotypes (Howcroft and Trauth, 2008; Bardzell, 2010; Marsden and Haag, 2016). In doing so, it is important to acknowledge that designs and practices are power-laden and culturally located (Irani et al., 2010), and take inspiration from various approaches reminding us to examine and hopefully address central feminist commitments of, for instance, empowerment and diversity—in the design and evaluation of systems (Bardzell, 2010)—by considering from whose angle or perspective collaborative work is considered (Muller, 2011).

5.1 Empirical Studies

In order to investigate global collaboration in GSD from as many different perspectives as possible, it was important to include several companies in this research. I will now introduce the three empirical studies that together inform this

dissertation. The companies that were chosen for these studies are among the largest IT companies in Denmark and, while they are all engaged in offshore outsourcing, they operate in various business domains and their customers vary from internal to external private or public customers.

5.1.1 Empirical Study 1

The first study involves an IT division within a large Danish bank (anonymously referred to as *BankIT* and *ScandiaBank* respectively). BankIT is considered one of the largest IT companies in Denmark and, at the time of investigation, it employed close to 21,000 people in Northern Europe. Since 2006, BankIT has been engaged in GSD by outsourcing development tasks to a Global IT vendor (anonymously referred to as *ITS*) engaging more than 750 people in India. Over the course of five months in 2012 and 2013, I conducted fieldwork with a fellow student in a BankIT department in Denmark and at their subsidiary development center at ITS in India.

5.1.2 Empirical Study 2

The second study was conducted in a large Danish IT company (anonymously referred to as *MData*) over the course of six months. MData has several branches in Denmark and, at the time of investigation, the company employed around 3,200 people. For more than 40 years, MData has delivered, developed, and administered IT for the Danish public sector; however, the company also develops IT solutions for private clients. One of their core competencies is SAP programming: a standard Enterprise Resource Planning (ERP) system used for managing a business. The field sites included MData in Denmark and their global IT supplier in India, anonymously called ITech. ITech is one of the large players in India offering IT services to a range of international customers. Since 2005, MData has been engaged in outsourcing IT projects to various IT vendors such as ITech in order to offer their customers competitive prices and to ensure extra resources and growth. In 2013, more than 200 people from five different global suppliers located in Poland and India collaborated with MData, and ITech was their largest supplier at that time.

5.1.3 Empirical Study 3

The third study involves a Danish IT service and software company (anonymously referred to as *Enterprise IT*). As I did a nine-month internship with Enterprise IT, this empirical study differs from the previous two as my role was to help and guide

the company in improving their globally distributed software development. I therefore had exclusive access to study the internal operations, strategies, and daily practices taking place in the company. In 2018, the company employed around 2,000 people at more than 15 different locations in Scandinavia. The company was founded back in the late 1970s and offers industry solutions, tools, and services for operational and administrative processes for the private sector; however, they also bid for public tender contracts from the public sector. In 2018, Enterprise IT was engaged in global offshore outsourcing in Poland, Ukraine, and India, and the fieldwork sites included in this dissertation involve three office locations in Denmark as well as an office location at their IT vendor in Poland, anonymously referred to as FutureTech.

Photo 1: Fieldwork photos from India, Poland and Denmark

5.2 Researcher's Journey

In the following I reflect on my journey as a researcher as I moved through the three different studies and developed as a researcher. I started out in an observant role; however, as I progressed through the different studies I began to take on a more interventionist research role (Zuiderent-Jerak and Bruun Jensen, 2007; Bjørn and Boulus, 2011; Bjørn and Boulus-Rødje, 2015), which included contributing to and guiding a company in applying new approaches while being in the field. I started out as a researcher with no personal stake (Walsham, 1995) in improving efficiency or productivity in their global collaboration. Instead, I found it important that my main purpose and focus were to first understand the particular complexities that participants in a distributed team are dealing with in their work. Due to my technical background as a software engineering student, I easily blended into the field; however, my academic training and minimal practical experience with corporate software work enabled me to maintain the role of the novice (Randall et al., 2010), albeit one equipped with technical language skills. Therefore, I was able to to maintain a distinct status as an outsider and thus, uphold a solid level of '*strangeness*' to the field (Neyland, 2007), which is important in order to investigate activities and uncover tacit assumptions, and to make them available for questioning and testing (Forsythe, 1999).

While it is extremely difficult to report on how or in what ways my role as a researcher has affected the fieldwork material, it is certain that the researcher cannot avoid influencing the field (Walsham, 1995; Mesman, 2007; Zuiderent-Jerak and Bruun Jensen, 2007). Thus it is useful to reflect on my role as interpretive researcher and how I tried to master the difficult task of accessing and understanding various perspectives and interpretations of the challenges involved in GSD work. In my effort to reflect upon my role as a researcher, I applied self-reflective approaches from action research (Zuiderent-Jerak and Bruun Jensen, 2007; Bjørn and Boulus, 2011) and maintained a fieldwork diary including reflective notes or video recordings of confessional statements in which I explained dilemmas, thoughts, or uncertainties in the field. These notes reveal some of the difficulties an empirical researcher may encounter when trying to understand, interact with, and communicate back to the field of study (Walsham, 1995; Mesman, 2007). The companies under study obviously hoped to receive some useful insights about their

globally distributed software development, and some practical advice on how to further their global efforts. Nevertheless, it is rarely the case that, during or immediately after the fieldwork, ethnographic investigations can offer unambiguous or actionable suggestions for how to improve a company's collaborative work. I recall a situation where my fellow researcher and I were leaving the field in India after one month of fieldwork and here the Danish head of the offshore subsidiary asked us what we then had found. Even though we continuously analyzed data while being in the field, this was a difficult question to answer as the preliminary findings needed time for further analysis, discussions, clarifications, and verification of the data. In future, therefore, when we left the field we were sure to inform the people involved in the study how they could get in touch with us and that we would return with more insights and present our results. For each of the three studies, I have taken time to both validate and present findings back to the participating companies.

As I continued on my empirical journey, I of course became more knowledgeable about the challenges and potential shortcomings present in the actual work and work practices in GSD. However, when shifting field sites, I was presented with different kinds of software work practices, technology stacks, tools, customers, and business domains, which kept my 'strangeness' to the field alive and prevented me from 'going native' in the field (Neyland, 2007). Moreover, I expanded my area of inquiry to include some of the local aspects that together form people's conditions for engaging in global work based upon their specific location, which in this case was the 'tech hub' of Bangalore. Not only did I start asking question about people's work hours, travels, and housing situations, but I also joined the local company bus as a way to explore the bodily experience of the daily commute for the majority of those working at the global IT vendor in India. In the morning and in the night, several company buses drove back and forth between the tech hub and the various residential areas in and around the city of Bangalore, and thus I joined the bus to ride and chat with people from the office heading in the same direction as me.

Moreover, I negotiated access with managers working specifically with global outsourcing, which enabled me to expand my research inquiry to include insights on the ways in which global outsourcing is implemented and supported from a

strategic point of view in a company. These insights are equally important for developing a nuanced perspective on GSD as it helped me understand the dynamic between those who are trying strategically to support and operationalize global outsourcing and those who have less say regarding the degree to which global outsourcing should be part of their everyday work.

The four papers included in this dissertation address various themes relevant for collaborative work in GSD, which I explain further in Section 0. However, there is an additional theme that has been with me throughout my research: the ubiquitous use of national cultural stereotypes (Kwek, 2003) in GSD. This is not to say that I do not acknowledge the presence of any cultural differences or the impact that multiple cultures may have on the collaborative work. Instead, what I found troubling was the way national culture or ethnicity was used and enacted by the people in the field, as demonstrated in the following quote:

> *"The Polish have a more intellectual approach toward the task" (Interview, Solution architect, Denmark, 31.07.2013)*

Based on the above we can of course query what exactly an 'intellectual approach' entails and for whom. Moreover, we may assume that this statement reveals a lack of common ground between collaborators. The key point here, however, is that statements like the one above infer certain categories that tie together intellectual approaches and nationality, while offering little reflection on the contextual contingencies or the actual conditions for accomplishing the work (Ang, 2011). The essentialist belief that there exists a nationally determined cultural behavior that we can utilize to explain people's actions in relation to, for example, collaborative work is still widely applied in GSD. Sometimes this conviction comes across as racist or is expressed through insulting expressions that I found both uncomfortable to listen to and difficult to know how to tackle. First of all, these descriptions made me uncomfortable as I found that they represented a narrow-minded perception of other human beings, drawing on distortions of power from colonial times. Secondly, I did not know how to respond to these statements as I, on the one hand, wanted to maintain a nice atmosphere around my informants and keep them willing to share their thoughts, but, on the other hand, it felt wrong if I were to feign to agree.

In the beginning of my empirical work, I tried to leave generalizing and offensive statements about culture in the background as I was aware of my necessity to build trust with my informants and, thus, I just listened to what people had to say. Later on in my empirical journey—and as I found myself accustomed to hearing the same benchmarking of, for instance, Indian consultants and their work attitudes—I started digging further into these statements by asking people to explain further or provide examples that would back up their benchmarking. What was interesting here was that when I started challenging these headlines of how national cultural behaviors impacted the work, the elaborate story would often be less biased and involve additional details with more nuance. In fact, when unfolding generalizing statements, these stories would often reveal inadequacies in the way the collaborative work and tasks had been assigned and distributed among collaborators or the degree to which expertise had been available for supporting work tasks that relied heavily on specific business knowledge (Matthiesen and Bjørn, 2017). In this way, culture helped point out where to further explore issues and complexities in the distributed collaboration, and so my research inquiries that identified the use of cultural stereotypes came into existence as I went on exploring GSD work in the field.

The use of cultural stereotypes was not only a practice I discovered within my field sites, but also at the university. In particular, when I followed a mandatory introductory course to university pedagogy at the University of Copenhagen, I experienced being dragged into a "culture in class" session and was asked to do a country comparison of my students from, for example, China versus Portugal in order to better understand their approaches to studying and learning. As an example of how cultures differ across nationalities, the lecturer from the Department of Science Education explained that Chinese students have fewer moral concerns when it comes to plagiarism, which I was highly provoked by. Thus, I started a discussion in the plenum and when evaluating the lecturers I asked if they would consider changing the ways in which cultural differences were introduced. Fellow PhD students have later informed me that the course has toned down the use of cultural-stereotypical examples.

Over time, I became much more daring in relation to the topics that I chose to delve into. In particular, I found it interesting to further understand some of the various tensions I was presented with, including, for example, the Danish collaborators' perception of GSD as a strategy and GSD in practice involving remote collaborators. This also meant that along the way I encountered situations that I was unsure how to tackle. In the following, I share a snapshot from my confessional statements. These are the reflections I wrote down immediately after an interview:

> I ran into a major issue when I confronted the Danish project lead with the issues that some of the developers in the project feel that the use of global delivery [global software outsourcing] is running a certain discourse of exhibiting a great success. When I mentioned to the project lead that some people were not very positive about the global engagement, he showed surprise and started asking me questions about it in a slightly interrogating way. I felt it was very uncomfortable, and I am not sure if I did something terribly wrong [...] I'm really not looking forward to listening to this interview! (8/13/2013, confessional statement)

Prior to these reflections, I conducted an interview with the project lead in one of the IT projects that I had been following for about a month at the time. During this interview we talked about some of the issues that I had encountered so far and I mentioned, for example, that several project participants had expressed to me that top management had decided that global outsourcing was a success and not to be criticized. While extremely aware that I, as researcher, needed to be careful and highly sensitive in conveying such tensions in a way that would protect my informants' anonymity, I found myself in a situation where I felt forced into an unfavorable position in the conversation that followed. Intermittently, the meeting changed from professional conversations on learning from challenges toward discussions of who was against global work and who had said what, when, and where. Even though I never disclosed any names, I found myself in an awkward situation as the project leader revealed that he undoubtedly knew whom I referred to, and so the discussion suddenly turned into a minor interrogation of what I, as researcher, knew about the project and its participants. I later wrote in my notes:

> It is now 7 days since I did this interview, but I have not had the courage to listen to the interview before now; it makes my stomach go up and down. (8/20/2013, fieldwork notes)

In revisiting these particular notes and writing the paragraph above, I feel vulnerable. However, it is important to acknowledge the way these experiences contribute to the overall research process of understanding the interrelatedness of the tensions, perspectives, and interpretations present in the collaborative work within global outsourcing. Not only did situations like this one prompt me to constantly reflect upon the implicit challenges that collaborative arrangements may encounter, but it also helped form me as a researcher; a researcher who, on the one hand, wishes to take seriously and challenge the various concerns and issues involved in GSD work from different perspectives, while, on the other hand, is extremely aware of the importance of her informants' confidentiality. As a consequence of this episode and the considerations it prompted, I became more intent on trying to figure out how I could interpret and convey some of the implicit and at times silent challenges involved in global outsourcing, which is further accounted for in the four papers included in the dissertation.

One effect of my efforts to take seriously the challenges of working with global outsourcing was that I also began to feel that I had more to say, which explains why I took on a more active and interventionist role as a researcher as I continued on my empirical journey. In the third and final empirical study, I applied interventionist approaches from action research (Zuiderent-Jerak and Bruun Jensen, 2007; Bjørn and Boulus, 2011) to initiate new strategies for dealing with everyday issues and challenges emerging within globally distributed software development. During my nine-month internship at Enterprise IT, I had exclusive access to study the internal operations, strategies, and daily practices taking place in the company, which enabled me to collect data through daily observations of numerous activities spanning from daily scrum meetings and project kick-off visits in Warsaw, to discussions of operational approaches toward realizing the company's overall strategy and corporate vision in relation to software development and outsourcing. In this context I was to assist a global delivery (GD) department in following, supporting, and supervising people and projects involved in globally distributed software work with their various IT partners located in Ukraine, India, and Poland.

In order to do so, I carefully examined and identified the challenges involved in the everyday practices of the ways in which the global collaborative work was

structured and carried out. As an approach to adjust, tweak, or intervene with the field, I constantly made suggestions on how to incorporate small practice-related approaches to the current global collaboration practices. Thus, I drove forward new initiatives and produced supporting artifacts or low-level tools that could help support the people in Enterprise IT who were already involved in or wanted to start up projects with GD. These artifacts included: hands on guidelines, GD vision communication, assessment models, a governance model, and best practice descriptions on how to engage in globally distributed collaboration on a daily basis. In this work, it became a mission to make sure none of the above-mentioned artifacts were exhibiting or reinforcing the status quo of well-known national cultural stereotypes. Artifacts were constantly reviewed and evaluated by both the practitioners involved (such as architects, managers, developers, testers, etc.) and by fellow colleagues at the university. In addition to that and as part of my interventionist approach I proactively developed and facilitated a workshop in Enterprise IT with assistance from the GD VP and a coordinator from the human resources department. This involved targeting employees working from Denmark and collaborating with software developers and testers working from locations situated in countries such as India, Ukraine, and Poland. The main idea of the workshops was to provide the employees with an additional lens and framework that they could use when observing surprising behavior and interpreting cross-cultural collaboration. In particular, I wanted to see if I could broaden the practitioners' understanding of geographically distributed collaboration by introducing the concept of implicit bias and discussing the implicit preferences and biases that exist when collaborating with remote colleagues. Through follow up questionnaires I evaluated and reflected upon the extent to which this new lens had had an impact. Finally, I presented my findings back to the top and middle management, and the company's global IT vendors from Ukraine, India, and Poland at a global outsourcing event in the company.

5.3 Data Collection

Between 2012 and 2018, I followed and collected empirical data from a range of different IT projects by closely observing the collaborative processes that took place at various locations within the three Danish IT companies their partners in India or Poland (see **Table 1: Overview of the data collection**). I followed various globally

distributed IT projects, interacted with project participants, asked reflective questions, carefully wrote down field notes, and captured screenshots for all kinds of collaborative work activities that I took part in. These activities included, for instance, kick-off meetings, knowledge-sharing meetings, progress and project management meetings, IT system demonstration sessions, and various scrum ceremonies such as daily scrums or sprint retrospective meetings. Data collection included a range of different research activities such as semi-structured interviews with people from all managerial levels and observations of the daily work of project participants such as IT developers, testers, or business specialists (see **Table 2: Interviews and observations**). All recorded interviews were transcribed verbatim and field notes were transferred and further typed up into digital versions, which made it easier to search through, categorize, and cluster them for further analysis. I also collected and analyzed various types of corporate documents representing organizational structures, system diagrams and descriptions, project presentations and evaluations, strategic documents, and business case descriptions, just to mention a few (for more detail see **Table 4**).

In addition to these formal representations of the companies that I studied, I also paid attention to and noted down comments about the atmospheres surrounding and work arrangements within the physical environments to understand the ways in which working at a particular location could impose certain conditions on the work situation. This desire for information about the contextual surroundings of where the work and collaboration took place at the different sites was not only useful for me as a researcher but also for the collaborators involved. For instance, a developer emphasized the importance of getting to know more about the work situation at his collaborators' location:

> *"Well at some point we discovered that, yeah well [...] they [the collaborators in India] were not even able to run [virtual] emeetings from their own computers; instead, they had to go to a meeting room. And there they would then sit and connect to the emeetings. Well then I understand so much better why they were consistently connecting one to two minutes late. So there you see, there are many things that can provide us with answers that make us understand their situation a bit better" (Developer, interview, Denmark, 05/10/2012*

Due to quotes like the one above, I carefully made sure to collect as much data as possible about the contextual surroundings at the office locations that I visited. So, besides investigating the tools and practices for engaging in global collaboration, I also paid attention to the ways in which the work was enacted through the available local hardware such as the desktop computers, laptops, headphones, information screens, conference equipment, etc. Moreover, I noted down details on how lunch and coffee breaks took place as well as how and when people reached their offices. In addition to conducting fieldwork inside the company premises, I also started to look outside the office walls, as explained in the previous section. For example, I joined the means of transportation for the majority of the people working at the global IT vendor site in Bangalore and accepted dinner invitations and weekend outings with the people I studied. Not only were these various kinds of data important for the research, but I also found it useful as a way to engage with and give back to the field both in India and in Denmark. Thus, while I was doing fieldwork in India, I made sure to note down as many subtle details as possible, as well as capture informal photographic and video material that could help the Danish collaborators expand their contextual understanding of the collaborative settings within the GSD arrangement.

In all three empirical studies, I had some level of access to a corporate laptop, which enabled me to participate in online meetings as well as access various project resources and collaboration tools (see **Table 3**). The data collected within the different empirical studies reflect the different levels of engagement that I had in the projects. This means that I started in a more observational role and, as I came to the final empirical study, I played a more active role in the field (see **Table 5**). For instance, I assisted in running daily scrum or retrospective meetings when a manager was unable to attend and I was often asked to provide my input and feedback. In the capacity of assisting the GD vice president (VP) in the daily work of disseminating GD approaches, services, and support, I was regularly invited to attend various kinds of meetings that included stakeholders at all levels in the organization (including managers, directors, HR coordinators, developers, testers, architects, business unit leaders, and the CEO). I had two weekly status updates with the GD VP to share insights and news within the organization and to discuss preliminary findings, obstacles, and future focus areas. Data also includes video and

audio data, questionnaire data, and fieldwork notes from two workshops that I organized and conducted. The workshops lasted two hours each, and entire sessions were documented through field notes, as well as video and voice recordings.

I considered all activities that I took part in as fieldwork and, thus, I wrote field notes and asked reflective questions from the very first interactions that I had with the field, which for instance included negotiating access, discussing field sites, and presenting my research project and focus to relevant gatekeepers and project stakeholders. In addition to that, I engaged in various kinds of social interaction ranging from being invited to private dinner parties, eating lunch, and spending leisure time with project participants. Even though some informants would in fact make jokes by asking me if I had forgotten my little black notebook, I did not write down notes during these more informal gatherings but instead I often wrote down my reflections afterwards.

Along with the data collection, I continuously validated, analyzed, or revisited data through several discussions with the people and stakeholders involved in the different companies that I studied. Moreover, engaging in discussions of the empirical data with my research group at the university paved the way for further empirical observations and interventionist approaches (Zuiderent-Jerak and Bruun Jensen, 2007; Bjørn and Boulus, 2011; Bjørn and Boulus-Rødje, 2015).

5.4 Data Sources

The empirical data comprises a fieldwork diary, confessional statements, observational notes, interview transcriptions, video recordings, project documents, internal communication (emails, slide presentations, IM conversations, etc.). Data also includes video and audio data, questionnaire data, and fieldwork notes from two workshops organized and conducted internally at Enterprise IT.

Table 1: Overview of the data collection

Empirical Study	**Study 1**	**Study 2**	**Study 3**
No. of field researchers	2	1	1
Length of study	5 months Oct 2012–Mar 2013	6 months June 2013–Dec 2013 + Empirical visits in 2014 + 2016	9 months May 2017–Jan 2018
Average time in field (hours/week) for the entire period	8.2	6.5	30
Presentation of findings in the company	1	1	1
no. of participants	25	2	60
Sites visited	1 India (Bangalore) 1 Denmark (Brabrand)	1 India (Bangalore) 1 Denmark (Ballerup)	2 Poland (Warsaw) 3 Denmark (Ballerup, Århus, Kolding)

Table 2: Interviews and observations

Empirical Study	Study 1	Study 2	Study 3
Interviews (no./hours)	12/26.2	21/16.4	4/2.25
Variety in roles of interviewees	Task manager, Developer, Task coordinator, Architect, Liaison officer, Business Developer, Deputy head, Senior development manager	Scrum Master, Developer, IT Architect, Solution Architect, Project Manager, Project Lead, Business Specialist, Delivery Manager, Tester, Translation manager, Global Delivery Expert, Global Delivery Manager,	HR Consultant, Developer, Vice President, Development Architect
Observations (hours)	80.3	58.4	218.5
Type of activity observed	Daily- and weekly Status Meetings, Kanban Meetings/eMeetings, Development Work, Clarification Meetings, Monthly Information Meeting at ITS,	Daily Scrum Meetings, Retrospective Meetings, Demo Sessions, Sprint Review Meetings, Weekly Project Meeting, Knowledge Management Meetings, development work, Informal and Ad hoc Meetings	Project Kick Off, Daily Scrum, Progress Meetings, Project Meetings, Retrospective, Demo Sessions, Estimation Meetings, Project Management Meetings, Software Engineering Workshop, Training, Technical Description- and Code Review Meetings

Table 3: Additional fieldwork activities and access

Empirical Study	**Study 1**	**Study 2**	**Study 3**
Confessional statements and reflective notes	✓	✓	✓
Fieldwork diary	✓	✓	✓
Research presentation	✓	✓	✓
Design and development			✓
Research intervention			✓
Corporate laptop	✓ (only in Denmark)	✓	✓
Office space*	✓	✓	✓
Document and artifact access			
Dedicated username/ email	✓	✓	✓
Subscription to relevant project/department email lists	✓	✓	✓
Online meeting	✓	✓	✓
Corporate SoME platform			✓
Collaborative platform (SharePoint etc.)		✓	✓
Source Code Management		✓	✓

*Office space indicates whether my fellow researcher and I were provided with a dedicated a desk space in the company, from where we could work, follow, and interact with the field during unplanned activities.

Table 4: Document and artifact analysis

Empirical Study	Study 1	Study 2	Study 3
Document and artifact analysis	✓	✓	✓
Document and artifact type examples	Global collaborative strategies, Department re- structuring descriptions, Employee overviews, Visitor's guide to India, Kanban board (physical), Kanban board, Speak your mind mail box, etc.	Culture & Mindset Report, GD & Information management, GD supplier governance Models, Weekly informal Project meeting presentations, Business case-, Interface-, Functions-, Process-, and Architecture-descriptions, Internal news letters, Scrum boards (digital), Requirement specifications, Project evaluation, Burn down charts, Sprint backlogs, Product backlog item (PBI), IT match making, GD Engagement overviews, etc.	Scrum board, Sprint backlog, Product backlog, PBI Technical description documents (TDD), Functional requirement and design documents (FRDD), FRDD estimation, Testing fundamentals-, Development methodology-, and Onboarding descriptions, Responsibility assignment matrix models (RACI), Governance models, Retrospective notes, etc.

Table 5: Interventionist research activities in study 3

Empirical Study no.3	
Activities	Implicit bias workshops, Questionnaires, Informal team meetings, Estimation meetings, Retrospective meetings, Supervising new projects with GD collaboration
Document and artifact production (Design and development)	Practice-based governance model for GD, Practice workflow descriptions, Introduction plan for new GD consultants, GD team structure presentation, Guidebook to GD collaboration and work: distribution of work, utilization of GD, coordination and communication, time registration, and collaboration culture in GD

6 Paper Positioning

The aim of this chapter is to position my research contributions within the larger research area, and to outline how my main findings support the overall contribution of my dissertation. Each paper is discussed in turn.

6.1 Paper 1: "Figure Out How to Code with the Hands of Others": Recognizing Cultural Blind Spots in Global Software Development

Despite the large body of knowledge accumulated on GSD work practices and processes and the various tools developed for supporting collaboration and mediating communication, the starting point for this study was to explore why distributed collaborators are still facing huge problems when trying to develop software systems across temporal and spatial boundaries. In order to understand why it is so difficult to "code with the hands of others," we conducted an extensive ethnographic study where we investigated the collaborative work carried out between a large Indian IT vendor and a Danish IT company. Even though the Danish company had seven years of experience in collaborating with their Indian IT vendor, there were several challenges present in the GSD setup. In particular, the challenges in relation to coordination and communication created tensions between distributed colleagues, which allowed the developers in Denmark to think of the global collaboration as a hindrance rather than an arrangement adding value to the daily development work.

We identified three key challenges for working in a globally distributed setup. One of the challenges was that the developers in Denmark would think of inquiries from their remote counterparts as interruptions or obstacles, in comparison to when a local colleague stopped by their offices; as a result of this, the remote colleagues in India were often ignored. Another challenge was that the Danish IT company focused solely on prepping the collaborators in India to collaborate with developers in Denmark, while paying little attention to adjustments that the local practices at the Danish office had to undergo in the global arrangement. For the developers in Denmark, new practices had been introduced by the management, which fundamentally changed their daily work practices and the nature of what was

otherwise considered as core development work. These changing practices involved increased communication and coordination with remote colleagues, which were activities experienced as "extra work" and not part of the Danish developers' job descriptions. Minimal credit was given to the Danish developers who now had to spent more time coordinating and communicating instead of writing code. Finally, we identified challenges in the way the work and task progressions were supposedly coordinated through daily coordinative artifacts such as a Kanban board. Not only was this Kanban board only physically available at the Danish office, but also it failed to include information about progression and need for support for the collaborators in India. While all the IT workers from the Danish site were listed at the board, the IT workers in India were consolidated into a single workforce labeled as "India," leaving the individual collaborators' work and task progression opaque.

The challenges presented above we found emerged due to various *blinds spots* that the people in the Danish company had regarding their own practices and the extra work and effort required for integrating coordination of the remote work. The way the collaborators in Denmark responded to or merely chose not to respond to their distributed colleagues' needs reveals that the Danish IT workers had blind spots in terms of what should be regarded part of the daily communication and collaborative practices with their collaborators at the Indian site. Moreover, the Danish IT workers' blind spots also became salient in the use of the coordinative artifact applied. Finally, we found that the management had blind spots for the implications that the substantial changes had brought along for the local practices at the Danish site.

6.2 Paper 2: Why Replacing Legacy Systems is So Hard in Global Software Development: An Information Infrastructure Perspective

Even if we acknowledge that GSD work often entails knowledge-heavy labor that does not easily transcend boundaries, it remains a challenge for research to figure out how software work can be divided, performed, and supported as a collaborative activity. Moreover, when looking at software engineering as a discipline, there are various research challenges in handling the multiple dependencies between system modules as well as the development methodologies and expertise embedded. In this second study, we set out to understand the work involved in designing, developing,

and implementing a large-scale and complex system such as a governmental IT system within a GSD setup. We conducted another ethnographic study in a large Danish IT company that was engaged in a GSD arrangement that included IT consultants working out of India. We focused on the nature of the system being developed by following a project to replace a 30-year-old legacy system for supporting Danish welfare benefits in Denmark. We applied the framework of information infrastructures to explore the complex and intertwined work involved in replacing and connecting components to an existing governmental IT system landscape. This led us to challenge one of the core assumptions in software engineering that acts as a main driver for engaging in GSD work, namely, the idea that we can reduce the need for communication and coordination if development tasks consist of pre- and well-defined interface descriptions (an assumption that was also present in this study of GSD). Here, the tasks being outsourced were considered technically suitable as they did not include translating hardcore Danish social welfare legislation into code, but instead consisted of integrating code to the existing code components in the social welfare system suite. In the ideal world, the integrations that needed to be done here would have followed predefined specifications. Unfortunately, in this particular case it turned out that the development tasks involved in the global collaboration never reached deployment.

6.3 Paper 3: When the Distribution of Tasks and Skills is Fundamentally Problematic: A Failure Story from Global Software Outsourcing

Software projects run the risk of failing due to various reasons and at various points in time during the phase of development, deployment, or in its final usage. For researchers, it is an exciting challenge to understand the multiplicity of reasons for why projects fail; however, gaining access to study failure cases is often difficult. Thus, coming across a case where things go wrong is a unique opportunity to further investigate the failure of a GSD project. In this third paper, we revisited the case from the previous paper (Matthiesen and Bjørn, 2015) to further understand how the development work within a GSD setup could go wrong without the management noticing it.

We were particularly intrigued in following the invisible dimensions of the collaborative work, which had gone under the radar only to be discovered too late. In order to investigate this, we explored the key coordinative artifact that was used in the everyday collaboration between the Danish office and the Indian office. This included the sprint backlog, which was used daily during the online meetings in the collaboration. Within this tool we found that the categories embedded and practices for monitoring or coordinating the work mattered for what and when problems were discovered. Investigating the sprint backlog and the practices around it revealed that little time was spent on informing or exchanging information about the actual work required while, instead, the speed of the development was prioritized. In practice, this meant that the collaborators from India and Denmark would connect through Skype for a 15-minute scrum meeting using a sprint backlog as their main coordinative artifact in the collaboration. The sprint backlog was presented as a simple spreadsheet that listed all ongoing, closed, and upcoming development tasks for the current sprint. During the online meetings, the scrum master would go through all the ongoing tasks with the aim of recording the overall progress in the sprint. The practices that were supported through the tool focused on the time estimates for how fast they could finish the individual software tasks, while little time was spent on articulation work (i.e., informing or exchanging information about the concrete work required). This approach had some side effects as the output of this focus produced burn down charts that visualized the speed of the development while complex aspects and obstacles in the software development work became residual. As a consequence, the software deliveries were sometimes registered as finished even though they were defective. In fact, as the project came to an end, it turned out that the deliverables developed in India had been faulty from the project's early stages. Interestingly, however, these problems were not openly discussed and instead some collaborators had been asked to keep their critiques to themselves. Finally, we found that the support provided from the Danish office had been lacking expertise for mediating and translating business knowledge into system-related goals for the collaborators in India.

6.4 Paper 4: Let's Look Outside the Office: An Analytical Lens for Unpacking Collaborative Relationships in Global Work

As the considerations on cross-cultural challenges in the collaboration have been and still appear as inevitable dimensions difficult to eschew due to their legacy in the literature on cross-cultural collaboration, research has yet to come up with analytical directions for how to understand or explain collaborative clashes as they unfold within the local practices in GSD. While critical reflections on cross-cultural collaboration as well as research into the power dynamics within distributed collaboration add to our understanding of the complexities of GSD work, there are still dimensions related to the nature of the local practices that remain unexplored. Instead, GSD work is often considered as a time- and space-flexible phenomenon where collaborative work transcends boundaries due to today's technology-enabled globalization. Global work as we know it, including the global work involved in software development, is often thought of as work that can take place *anywhere* and *anytime*.

In the fourth and final paper included in this dissertation, we challenged some of the core assumptions about GSD that relate to the flexibility of engaging in global work. Our research took initial inspiration from a statement offered by a developer working out of India. This developer expressed that the difficulties of engaging in global work depended on one's physical location, and this inspired us to explore the bodily experiences of the various physical surroundings in which the global work is embedded. We investigated the spaces where GSD work would normally take place, such as the offices in Denmark and India, but we also looked outside of the office to explore the more subtle infrastructural circumstances, which, for instance, involved daily commutes, and housing possibilities.

The study revealed that daily commutes, housing possibilities, domestic obligations, technology availability, and flexibility in work hours all posed certain challenges for the ways in which people were able to participate in global work. In particular, we found that it matters greatly whether you are a developer working in India or a developer working in Denmark, due to the infrastructural differences found at the various locations. These infrastructural differences exist both inside and outside the office walls but come together as the overall conditions for engaging in global work.

For instance, the politics of how time is managed look very different depending on whether you are working from Denmark or India. For the Danish developers, work can take place both at the home and at the office, and this enables work to be much more flexible (but also all-pervading) in relation to caring for a family: work can be split up through the day to ensure, for example, that the children can be collected from kindergarten in due time for preparing dinner. This flexibility is not present at the Indian office, where only desktop computers are available at the office due to various data security measures. For the same reason, there were core differences in the ways in which people organized their living situations in order to attend to nine and a half hours of work in India, or eight hours in Denmark. And, in order to skip the traffic congestion in the streets of Bangalore, some developers even chose to stay close to the office during weekdays and to only return to the family residence on weekends. This model is far from how people in Denmark would typically organize themselves and their families.

In the same way, the politics of place also posed certain asymmetries in the conditions for how people engaged in a global collaboration, as both power and knowledge mainly resided at the Danish site. In particular, we discovered how the rotation plans of temporally placing collaborators from India at the Danish site were formally structured but unevenly practiced. This created a level of uncertainty for the developers who were still waiting for the opportunity to go abroad while those already working at the Danish site would raise their individual value in the project.

7 GSD Stereotypes

Prior empirical observations of collaborative work in GSD have found a common practice among the IT professionals involved in these cooperative arrangements, namely, a tendency to explain issues through cultural rhetorical constructs (Jensen and Nardi, 2014). The findings from the four research papers suggest that this common way of explaining collaborative issues was particularly manifested—by the IT professionals working out of Denmark—through negative cultural stereotypes of their remote and foreign collaborators, which shaped the collaboration in problematic ways. Therefore, I set out to identify the various coordination and communication issues that were commonly explained with reference to cultural differences. Building on the findings from the four papers, I develop here a **GSD Stereotype Framework** that describes ***where*** we should go look for issues and ***what*** we should attend to in order to explore beneath the surface of various stereotypes and to further understand the concrete coordination and communication challenges in GSD. Furthermore, I expand on these findings by bringing in ***implicit bias*** as an analytical lens to unfold what implicitly becomes part of our common language, and the ways in which we can move forward and develop a more suitable vocabulary within cooperative arrangements in GSD.

The GSD Stereotype Framework focuses on the following three areas: 1) *the organization of work*; 2) *the collaborative technologies and system structures*; and 3) *the conditions for work*. I now turn to each area and introduce some key attributes, which I identify as useful for explaining what underlies the creation of stereotypical arguments in GSD.

7.1 The Organization of Work

Regarding this first area, the way the collaborative practices are organized builds upon a range of *prejudices* among distributed collaborators and about software work as a distributed collaborative activity. When inquiries from collaborators working out of India are ignored and interpreted as interruptions instead of efforts of collaboration (Matthiesen et al., 2014), these kinds of 'practices' demonstrate some of the prejudices that collaborators hold about other collaborators and that become manifested in the way the work is organized through tools and practices. In the example with the physical Kanban board in Denmark, we saw how the individual

collaborators working out of India were only collectively represented at the board level, and information about their work was often passed on by one of the few collaborators at the Danish office who were most engaged and thus informed about the work performed at the Indian site. This organization of work also highlights the inherent *power hierarchies* that emerge within the cooperative work arrangement of hiring an external vendor (Bjørn et al., 2017) that is apparently expected to figure things out with minimal effort or involvement from the client (Matthiesen and Bjørn, 2017).

Based on the *preconceptions* about what were and were not suitable tasks to be completed at a remote location, the developers working out of India were assigned low-status software tasks—assumed to be straightforward and 'technically suitable'—for GSD work (Matthiesen and Bjørn, 2015; 2017). Clearly there was a preconception that considered the interface tasks involved in replacing a governmental legacy system as a pure technical matter. However, these 'technical' tasks were deeply rooted in the installed base of socio-cultural and socio-technical structures as well as the subtle yet undocumented implementations of welfare legislation. For example, the technical tasks of plugging into different systems through technical interfaces involved unforeseen efforts of excavating knowledge about the given interfaces from various systems (e.g., internal, third-party external, or legacy systems) and people outside the project. This proved to be a challenging mission as these extended efforts and the expertise required for solving these tasks in the GSD setup were not accounted for in the everyday work practices or in the organization of work and workers (Matthiesen and Bjørn, 2017). In retrospect, it is clear that the division of work was not only unfortunate and inappropriate given the nature of the system (Matthiesen and Bjørn, 2015), but, also, it turned out that the initial division of tasks had presumably been performed based on the Danish office's preferences regarding what was considered high-/low-status work instead of careful consideration for establishing the best possible conditions for distributed collaboration.

As I already described with the example of the underwhelming representation of global collaborators at the Kanban board, this inexpedient division of work tasks that favored the project participants working from the Danish office similarly

relates to the power hierarchy (Hinds and McGrath, 2006; Hinds et al., 2015) of the client-vendor relationship (Bjørn et al., 2017). Moreover, this division of tasks reveals a demonstration of power, which arises upon the economic drivers of GSD, and where "cheap labor" becomes juxtaposed with low-quality and underperforming labor (Metiu, 2006; Vora, 2015; Matthiesen and Bjørn, 2017).

While collaboration-related challenges in GSD are often explained as the outcomes of cultural differences, these findings remind us that we need to search deeper in order to arrive at the crux of the matter. Thus, when we seek to ethnographically explore collaborative work practices in their 'natural settings' and take the collaborators' perspective (Blomberg and Karasti, 2013), it is crucial that we pay attention to the underlying attributes: *the power hierarchies, prejudices,* and *the preconceptions*—that collaborators have among each other or even against the work they are presented to—which serve as input for the ways in which the work is organized and performed in practice.

7.2 The Collaborative Technologies and System Structures

The second area that both researchers and practitioners should focus on concerns the collaborative tools and systems structures. On this issue, I earlier stressed how the *categories* embedded within the tools supporting collaboration in GSD matter for the ways in which the collaborative work is portrayed (Matthiesen et al., 2014; Matthiesen and Bjørn, 2017). In the example with the collaborators who were consolidated into one entity row called "India," the work efforts of the individual collaborators were obscured. In this way, the categorization schemes embedded within the collaborative technologies have the ability to direct attention toward certain traits of the collaborative work, and to overlook other aspects. This was the case with the sprint backlog and burn down charts, which directed attention to unfortunate results of timely progression while critical aspects about the code quality or the need for knowledge support were rendered invisible (Matthiesen and Bjørn, 2017). Not only were the inappropriate division of tasks among collaborators problematic when not accounted for within the applied work practices, but also it turned into an even greater issue due to the lack of feedback that the tool provided. In particular, the preconception of pure technical tasks suggested that certain quantifiable categories had been embedded into the key coordinative artifacts of the

sprint backlog and the burn down charts, while qualitative measures were neglected.

Investigating categories within collaborative system design is not novel; in fact, the role and the politics of categories have been discussed since the inception of CSCW as a research field (Winograd and Flores, 1987; Suchman, 1993; Bowker and Star, 2000; Star and Bowker, 2007). Nevertheless, by paying attention to and making room for a sensibility to understand the categories that the tools in GSD come from, we may be able to make better sense of *what* or *whom* it is that the technologies neglect or promote in a global collaboration. However, when reflecting upon the categories that implicitly go into the technologies we build, we also have to ask ourselves: *from where* do these categories arrive? In answering this, we not only need to identify how categories are reflected in the design, selection, and appropriation of the particular technological artifacts, but we also need to consider how these categories relate to the certain power hierarchies and preconceptions that are present in GSD.

7.3 Conditions for Work

Refraining from reinstating strong narratives about cultural behavioral patterns and how these are breaking collaboration, the third and final area in GSD Stereotype Framework concerns the conditions for work. This area I develop based upon my findings showing that global collaborative work is intrinsically tied to people's physical location and, thus, the circumstances and the infrastructural aspects involved (Matthiesen and Bjørn, 2016). For instance, hardware access, daily commutes, housing possibilities, or travel/rotation policies are just a few of the features that make up the conditions for participating in GSD work. Nevertheless, these features are rarely included in the analysis when trying to understand the complexities of transnational collaboration (Matthiesen and Bjørn, 2016). Thus, we need to expand our purview to pay attention to the way conditions for engaging in global work differ due to the many variations of working across sites. Although we may not be aware of, or consider important, the conditions for work that are available at the different locations, our interpretations of collaborators' work and work efforts risk being misinterpreted or affected by various *misconceptions* that help confirm existing stereotypical beliefs.

One key issue in all of this is access (or otherwise) to hardware; but, even when the hardware seems to be in place, it may turn out that certain specifications constrain the ability to participate in global work on equal terms. I previously described how the conditions for participating in virtual conference meetings (emeetings) required additional efforts by those working out of an Indian office. In particular, the people in India had to leave their desktop computers to go and log on to another computer located in a meeting room. Initially, the Danish collaborators were unaware of these additional efforts and so the India-based colleagues (virtually) arriving a few minutes late was met with little empathy by those in Denmark until they knew more about the physical constraints that the developers in India had to deal with. The example demonstrates that, in order to establish the best possible conditions for collaboration in GSD, it is important that teams get acquainted with the actual office settings, conditions, and constraints of where the work will be performed in practice on an everyday basis. However, if these conditions for work remain uncharted, there is a risk that collaborators will develop *misconceptions* or lack understanding of their remote colleagues' behavior, which can have consequences for how the collaboration is interpreted.

In addition, the emergent dissimilarities in the conditions for engaging in a global collaboration also relate to how the politics of time and place is performed differently at the different sites as well as inside and outside a particular office. In particular, work looks different depending on whether or not you are in a (physical/hierarchical) position to negotiate when and where you work. Therefore, taking into consideration the ways in which certain power hierarchies are manifested in the conditions for work is again relevant here. For instance, your conditions for work differ depending on where the knowledge is located and when it is accessible to you (Faraj and Sproull, 2000; Avram, 2007). There is also a difference between whether you are allowed to take a day and work from home, or whether you have to obey to a (flexible) seven-and-a-half-hour or a (strict) nine-hour workday (Matthiesen and Bjørn, 2016).

7.4 Adding Implicit Bias to the GSD Stereotype Framework

In the above, I have identified the three GSD Stereotype Framework areas that are important to consider in the effort to move beyond explaining collaboration-related

challenges in GSD with reference to "cultural differences" through negative stereotypes. The boundaries between the three GSD Stereotype Framework areas are not strictly defined. Instead, these areas intersect and correlate as tools or conditions for work may insist on certain organization of work or vice versa. Nonetheless, identifying wherein challenges arise is only part of the task here. To take concrete action to move beyond negative stereotyping, it remains important to explain *why* people apply negative stereotypes to explain inappropriate organization of work, insufficient collaboration tools, or incoherent conditions for work. When presenting the GSD Stereotype Framework, I pointed out power hierarchies, prejudices, preconceptions, and misconceptions as core *attributes* that move us closer to recognizing *what creates* the various stereotypes that implicitly become part of our common language and, thus, go into the design of the system structures and technologies that we apply in GSD. I now refine the GSD Stereotype Framework attributes, which guide us to become more aware of *what* in the collaborative work contributes to the creation and re-inscription of, for example, negative national cultural stereotypes.

As for the use of national cultural stereotypes, scholars have explained this in terms of the easy and mainstream applicability of essentialist cultural frameworks (Kwek, 2003). It has also been argued that culture may serve as a power-laden rhetorical strategy to maintain one's hierarchical position in a company (Jensen and Nardi, 2014). Relatedly, a focus on cultural blind spots may help in identifying where people have their focus and thus, what become their blind spots (Matthiesen et al., 2014). However, the analytical concept of cultural blind spots does not help us to understand the mechanisms behind *why* these blind spots materialize as well as the selection of *what* initially becomes our focus (or not). I apply the concept of implicit bias (Greenwald et al., 1998) as an analytical means to further unpack why, for instance, blind spots appear in the first place or, more precisely, *what creates* the culture stereotypical arguments in GSD.

Implicit bias is an interesting analytical perspective as it describes how even the most well-intentioned people may operate with stereotyping and prejudiced behavior without this being part of their explicit agenda. By attending to the implicit biases present in GSD, we can refrain from the unhelpful, personal criticism

entailed in pointing out individuals' hostile or 'bad' behavior. Instead, we are able to learn more about the different views and perspectives that contribute to the range of problems emerging in the work and among collaborators in GSD. In this way, focusing on implicit bias serves a timely and alternative analytical approach toward understanding collaborative work within transnational encounters in relation to what produces the application of cultural stereotyping within the cooperative work arrangements in GSD.

However, ethnography differs epistemologically from psychology—in the sense that ethnographers do not believe that we can access or measure people's minds—so when we as CSCW researchers apply ethnographic research approaches, we need to study what people do in practice, in their natural settings, and use the specific language that is meaningful to those we study (Blomberg and Karasti, 2013). And so, you may find that, when I take a theoretical concept that has been developed based upon epistemological stances from psychology, I then break with some of the core ethnographic principles that I present in this dissertation and that I have carefully applied throughout my doctoral research. However, I apply implicit bias as an analytical strategy not to measure the degree to which people have biases against other people. Rather, I consider whether implicit bias can serve as an analytical lens that complements the GSD Stereotype Framework in addressing the actual coordination and communication issues in GSD as an approach toward moving beyond negative stereotyping.

In an attempt to make purposeful and culturally intelligent simplifications (Ang, 2011), I develop my analytical lens to move beyond cultural stereotyping in GSD. In doing so, the dissertation takes those GSD Stereotype Framework attributes that are hard to tell from each other and assembles these into an attribute of *implicit bias*. These are the attributes that cover what I have empirically interpreted and analyzed as the preconceptions and prejudices that people demonstrated or articulated about the work involved in GSD or toward their distributed and remote collaborators. In this way, implicit bias as it is applied here serves as an umbrella term, which includes the interpretations of national cultural behaviors and skillsets as well as the descriptions of the work involved in GSD. By restating implicit bias, the framework distances itself from trying to move into a person's head and judge whether a

collaborative decision, structure, or action emerges from a person's preconceptions or prejudices about the world she or he takes part in, and instead it describes the preferences that are implicit for that particular person.

What this means in practice, especially when paying attention to the organization of work through collaborative tools and practices, is that implicit bias manifests itself in the way that collaborators are disparaged, meetings are executed (locally/exclusively), or work tasks are supported, communicated, and distributed across sites. In this way, implicit bias is complementary to the inherent power hierarchies that come along when cooperative work arrangements are to thrive within the labor political tensions and economic incentives that drive forward global software offshore outsourcing in an organization. With regard to the collaborative technologies and system structures, implicit bias is likewise present and manifests itself along with or through the categories embedded in the collaborative tools and systems, which have the ability to hide or marginalize important aspects of the work or the people in the collaboration. For the third and final area (the conditions for participating in distributed work), implicit bias manifests itself in the way people interpret or misinterpret collaborators' work and work efforts due to the lack of information about the physical surroundings and infrastructural constraints or opportunities offered at a certain location. In particular, when there is a lack of information about the physical surroundings and societal and infrastructural aspects available at the different locations, certain biases and misconceptions may start to arise.

In ending this chapter, I refine the attributes in the GSD Stereotype Framework to concern: power hierarchies, categories, misconceptions, and implicit bias. Focusing on these attributes can guide us to become more sensitive toward *what* contributes to the creation and re-inscription of, for example, negative national cultural stereotypes. It would be naïve, however, to think that by applying this framework we are then able to remove cultural stereotyping in transnational cooperative work arrangements. Instead, the GSD Stereotype Framework offers a set of guiding principles that both researchers and practitioners can apply to move beyond applying stereotypes as explanations for collaboration issues, and to instead become more sensitive to the complex, intertwined, politically and locally situated

challenges and issues that emerge within cooperative work arrangements in GSD. This also means that I do not claim that the framework serves to *remove* stereotyping in GSD. Rather, the framework offers guidance on where to start dealing with the challenges, which otherwise risk being closed to further scrutiny due to narratives building upon various negative stereotypes in GSD. Nevertheless, as we become more aware of how challenges materialize in GSD—that is, through understanding the areas in which we should look, and the key attributes we should look for (illustrated in **Table 6**)—we are able to question the gross national cultural stereotypes as these can no longer function as proxies for the majority of the collaborative issues in GSD. These are issues that I have identified as emerging from:

- Inappropriate **organizations of work** due to the particular power hierarchies and implicit biases involved;
- Insufficient **collaboration tools and system structures** due to incomplete categorization schemes that favor certain groups of people based on their location or affiliation, or that favor progression over quality;
- Misconceptions and implicit biases about the **conditions for work**, which appear due to information deficiencies about the way the infrastructural aspects or physical surroundings at a certain location pose limitations and constraints for participating in the globally distributed collaboration.

Table 6: The GSD Stereotype Framework Areas and Attributes important to consider when unfolding collaborative challenges in GSD that risk being explained through various negative stereotypes.

Areas / Attributes	Organization of work	Collaborative technologies and system structures	Conditions for work
Power hierarchies	✸	✸	✸
Implicit biases	✸	✸	✸
Categories		✸	
Misconceptions			✸

8 Applying the GSD Stereotype Framework in Practice

In the previous chapter, I developed the GSD Stereotype Framework that describes areas and attributes important to consider when unfolding collaborative challenges in GSD that otherwise risk being explained through various negative stereotypes. The framework serves to capture the situated and real-life challenges and complexities involved in developing software across various (geographical, temporal, and/or cultural) discontinuities. As a novel contribution to CSCW, I introduced a focus on *implicit bias*, which I will now demonstrate serves as an interesting and important attribute that can point to some of the mechanisms that contribute to the use of cultural stereotyping in GSD.

The key to mitigating implicit bias's unintended effects (such as negative stereotyping) is first and foremost to recognize that all people have biases and, thus, it is essential to become aware of our own implicit biases (Stone and Moskowitz, 2011). While it may be painful to realize that we have implicit biases regarding different social, professional, or ethnic groups, it is nevertheless an important step toward doing something about our biases. The more we become aware of our implicit biases, the better we can combat the effects of these biases and, in this case, start work on improving the way we work together across geographies, cultures, professions, business units, and companies. For CSCW researchers, implicit bias is an important means of expanding our perspectives when analyzing and understanding the actual work practices and how these are or can be supported through technologies. Nevertheless, if we are to become more aware of our biases, the question is: how do we then go about doing this? Or, more importantly, what happens when we bring back considerations from the GSD Stereotype Framework to the field—as a strategy to address the challenges that stereotypes risk hiding within the cooperative work arrangements in GSD?

In the following, I lay out results from my third empirical study, where I—compared to the previous two empirical studies—took on a more interventionist research approach (Zuiderent-Jerak and Bruun Jensen, 2007; Bjørn and Boulus, 2011) as I could no longer ignore that the common vocabulary used in GSD had problematic side effects. To move forward, I highlight implicit bias as a way to address the use of

stereotyping and uncover the collaborative challenges and issues that are obscured by the current language discourse in GSD. Specifically, I present results from my engagement in Enterprise IT where I both explored and tried to bring to attention implicit bias within each of the three GSD Stereotype Framework areas defined in Section 7. Other areas of focus (i.e., the organization of work, the collaborative technologies and system structures, and the conditions for work) also serve as structures for discussing the challenges I found and the ways in which these can be addressed and potentially mitigated moving forward.

In the empirical field, I carefully considered the ways in which I tried to bring in a consideration of implicit bias to the real-life situational context of Enterprise IT. In practice, this meant that, in order to reduce the chances of producing material that reinforced additional stereotypes or reflected my own implicit biases, I discussed the various artifacts I developed with people in Enterprise IT as well as with colleagues at the university. Moreover, when recruiting for the workshop that I developed and conducted within the company, I did not want to advertise the workshop topic and focus as involving themes of "cultural stereotypes" and "raising attention to how we are biased," since this would risk that the workshop's aim could be misinterpreted as being about software developers' resistance toward global collaboration as a way to preserve their own careers and job security (Rost, 2004). Since this was not the intention, I made sure not to signal or imply any negative implications that people were biased or that the workshop was about 'curing' them if they signed up. Thus, the challenge was to develop a workshop where I could introduce the idea of implicit bias to the target groups—without implying that people are biased prior to signing up—in order to move the discussion from 'cross-cultural' challenges to a focus of the actual coordinative and communicative challenges involved in the GSD setup. Further details on the design of the workshop can be found in Appendix 10.1.

8.1 Stereotyping Within the Organization of Work

When exploring challenges within the organization of distributed work as it takes place through tools and work practices, I have—throughout my doctoral research—encountered how people tend to refer to their collaborators' work practices and behavior as a feature of their national origin. Thus, in my efforts to raise attention toward implicit bias, I tried during the workshop to take descriptions of cultural

differences to them in order to see how far these descriptions would go in relation to making up negative stereotypes. For instance, I asked the workshop participants to engage in a *personas* exercise and to discuss and develop descriptions of a typical global colleague persona based on their experiences with global collaboration.

Personas were first introduced in the late 1990s (Cooper, 1999) and have been widely applied in a range of product development activities (Nielsen et al., 2013; Nielsen and Storgaard Hansen, 2014; Hjalmarsson et al., 2015). While personas serve the purpose of representing the personal characteristics and goals of the different user types that might use a product, brand, or service, focusing on such fictitious, archetypal users also runs the risk of re-inscribing existing stereotypes (Bardzell, 2010; Rode, 2011; Marsden and Haag, 2016). The participants were asked to consider and create personas with attributes such as: nationality, role/function (developer, tester, etc.), skills and work experience, marital status and family, work attitude/behavior/habits/preferences, communicative skills, organization of work/coordination skills, conflict-handling skills, and collegial skills in relation to work and socializing/humor. The groups then presented their personas in a plenary. One of the groups described how a developer from India would be very hierarchically oriented by obeying the manager and doing exactly what he or she was asked to do and nothing more. According to one of the managers, the Indian collaborators would just say 'yes' even though they were not going to accomplish a task:

> *"[...] the Indians, they do not know the word 'no,' they just say 'yes' to everything, you just have to understand what the word 'yes' means, it does not always mean the same as when we [Danes] say 'yes.' When we [Danes] say 'yes,' it means we have understood what has been said; we agree and have understood the task. When the Indians say yes, it just means they have heard what has been said, but they have not committed to it."* ***(Senior Product Manager, 1st workshop)***

What became interesting here was that while the statement above was presented, another manager from the same group appeared uncomfortable with creating these general descriptions and said:

> *"... at the same time this also depends on the person"* ***(Resource Manager, 1st workshop)***

The resource manager quoted above said several times that these descriptions really differed from person to person and that they could not always make these kinds of generalizations. This unfolding of collaboration-related issues that were first described as being due to (very general descriptions of) how people work and act was exactly the purpose of the exercise. In particular, my mission was to have people reflect upon the degree to which these general descriptions reflect reality when further discussed in plenum. The manager who was uncomfortable with creating gross stereotypes added a story that supported the need for applying and articulating a more nuanced opinion on the individuals' prerequisites for accomplishing a given task or taking on a given function:

> *"[…] we have Poles that have been here [in Enterprise IT] for 12 years, and then when we're taking in someone from India, then there are people [in Denmark] who've got this kind of attitude like: 'this guy is so bad, he can't do anything,' and then I'm like: 'Yes, but you compare him with one of the Polish guys who has been here for 12 years now! If it had been a Dane, he would also need a little bit of assistance'"* ***(Resource manager, 1st workshop)***

The persona exercise demonstrates how easy it is to start a conversation about nationality. What is problematic about this is that when nationality keeps coming into play as a qualitative factor in relation to how skilled you are and how well you are able to engage in transnational collaboration, these stereotypical descriptions do little but serve as a way to confirm existing biases instead of adding useful guidance as to how we should try to engage in and establish common ground with globally distributed collaborators. However, what was interesting with this persona exercise was that this session continued with a discourse that bounced back and forth between explaining global collaborators' behaviors through stereotypical descriptions (as I had encouraged them to do with the personas exercise) and elaborate anecdotes that would demonstrate a counterargument. In particular, as soon as one person challenged a general description or a stereotype of a person (that would otherwise be explained by 'national cultural differences'), the group would then support the new discourse and start looking into the nuances with additional explanations of a given situation by saying something like: *"This situation you would also witness with our own people in Enterprise IT."*

Before the second workshop, I asked the participants to reflect upon whether they saw any value in boosting their attention toward implicit bias, and thus, the starting point for the second workshop was to have the participants discuss and reflect upon implicit bias in relation to: 1) their own or their colleagues' work and practices; 2) the efforts and engagements they contributed themselves; 3) the contributions from Enterprise IT and the GD group; and 4) the tools they used in their everyday work with GD. One of the participants explained, how he—after the first workshop—had gone back and constantly tried to articulate all the examples he encountered of implicit biases or, in his words, the *prejudices* he met with in the teams' everyday interactions:

> *"[...] most of all I see it as prejudice; there are a lot of prejudices about ... especially Indians and as said previously about [software] development, but to try in the teams that I am a part of, to articulate these things here, because there are many of these teams that are not used to working with Indians. And there are some things, some prejudices that—unfortunately—are right, but there are also other things where there are pure prejudices"* ***(Functional Architect, 2nd workshop)***

The quote above demonstrates the challenges that exist in improving the way we sort out and navigate between the truths, myths, prejudices, and implicit biases that we have about each other. In particular, according to the functional architect, there are biases and prejudices inherently built into the ways in which his teams think about work that involves collaboration across countries. This was also evident in several of the discussions such as the one below where the participants discussed how they often experienced a tendency to develop certain stereotypical narratives about their globally distributed colleagues:

> ***Resource Manager:*** *"Yes, and then it can turn into a whole culture just talking in that way, and we've experienced a little of the same, where I've been sitting and being a bit like: 'now you STOP.'"*
>
> ***GD VP:*** *"Yes, something like: 'Indians never say "no"!'"*
>
> ***Resource Manager:*** *"Exactly. And then when they do say 'no,' then they are just really annoying. It's just the way you talk about them [the Indians]"*

> ***Senior Product Manager:*** *"It's also much easier to say: 'it's also just because he's Indian!' It is a little bit harder when it is a colleague—sitting next to you—who makes the same mistake. Then you know: 'Well, that's because he had not been told that's how we do this and this,' right?"*
>
> ***Resource Manager:*** *"Yes, it is a bit of the same I'm dealing with, this idea that you cannot just take in an Indian [GD consultant] and then they're just up and running from day one. And well, hey! The same goes for a new Danish employee, right?"*

In the above conversation, the managers express how there is a tendency to grow a cultural practice around building stereotypical descriptions and prejudices in the company, and one example was related to the stereotypical description of the Indians' behavior and proficiency. However, this cultural practice of reinforcing existing stereotypes about the people we believe are different from ourselves is difficult to combat. As the manager explains, the discourse, where stereotypical descriptions are cultivated, can easily get out of control, and this is where concepts such as implicit bias can help us articulate, reflect upon, and combat negative cultural practices. The importance of having such analytical concepts became clear to me, as it paved the way for constructive conversations about what the participants found problematic when working with GD without stigmatizing anyone as a bad, racist, or uncooperative person. In particular, the attention toward implicit bias not only points out how these cultural practices bring along different expectations and appreciations of a collaborator depending on whether the collaborator is from the same or a different country, but it also reveals how these expectations pose distinct preconditions for carrying out the work depending on where in the world the collaborator is located. This is a subject I will return to later in this chapter.

One of the discontinuities that have also been with me throughout my research concerns the use of Danish language despite the fact that English is officially the corporate language in Enterprise IT. According to several managers, the use of English language is not implemented as a common practice. Below, a manager describes how this is problematic for the work practices that take place:

> *"I was talking to somebody from [a department in Enterprise IT]: Maria. They have run a so-called 'Danish' scrum meeting and then afterwards there was a team member who then called the Poles and told them what had been said [during the meeting]. And here they noticed that they [the Polish consultants] were a bit disappointed when they were not included in the meetings… 'So, maybe you should just switch to English, because you're not able to report everything that happens [during a meeting]!' Come on!"* ***(Resource Manager, 2nd Workshop)***

When the manager tells this story, she is really annoyed by the way people tend to engage in collaboration with their global partners and colleagues, and to her it is quite obvious that a lot of content and context is lost with a practice like the one described above. Language indeed plays a huge role, and to several of the managers this really was a source of frustration:

> *"I have some frustrations… because there are some in my team that we collaborate with, and they do not understand that they have to do things in English … 'but developers must be able to read what you write,' 'no no no, they do not have to see what I am doing as a business analyst,' 'oh yes they do, they have to do code based on your descriptions, so they [the developers] must be able to read what it is in fact they are suppose to code.' Well, we have many discussions like this one!"* ***(Functional Architect, 2nd Workshop)***

There may be many reasons why people keep using their native language instead of the official corporate language. For instance, in the first empirical study, one reason was that the staff in Denmark did not feel comfortable speaking English:

> *"Then there is all that with the culture and language, and so on. And it's a challenge; it really is, you know, a barrier. And again like ScandiaBank is a super tanker that needs to get used to turning the whole ship to speak English, then it is a super tanker with the staff too. Well, we are generally reasonably good at English in Denmark, but we do not believe it anyway […]"* ***(10.05.2012, interview, developer, Denmark)***

The lack of confidence that non-native English speakers deal with is an important factor that companies need to take seriously (Neeley et al., 2012). Nonetheless, if we consider these work practices in the light of implicit bias, these may also indicate an implicit bias that underestimates the importance of what foreign and remote collaborators need to know in order to do their job. In this way, there are various

implicit biases among the different professional roles in relation to who should engage in global work and, for instance, who should make adjustments to the ways in which business analysis work is executed and results are handed over in a globally distributed project. This is also apparent in the following quote:

> *"[…] There are many things that are delivered in Danish; for example, project or product summaries and overviews. It's a bit like: 'Ahhh come on, honestly, can't you just do it in English?' What I'm experiencing is a mentality saying: 'Can't you just pick out the things that are important for the Indians to know?'— 'No, why can't you just do it in English from the start?'"* ***(Resource Manager, 2nd Workshop)***

Again, the quote not only illustrates a common issue around having people change their habits and adopt and use the official corporate language. It also indicates how software work or globally distributed software work is thought of, namely, that there are some implicit biases that underestimate the intellectual challenges of developing software that requires a lot of information and knowledge (which I also demonstrate in (Matthiesen and Bjørn, 2015)) that reaches beyond technical instructions that are stripped of any contextual or situational content. When someone is asked to hand over information from, for example, a scrum meeting, it is inevitable that the risk of losing a lot of important information is high. In addition, the preconception about what software work requires in relation to skills, knowledge, and support I also discuss in an earlier study (Matthiesen and Bjørn, 2017) where the distribution of tasks and skills proved highly inexpedient. Now in the quote below a manager further unfolds what he finds problematic about the way in which IT developers' work is considered by the management:

> *"[…] the funny thing is that there is also this bias, which typically comes from the management, saying that 'you "just" need a programmer, because they just need to do code' […] 'Okay, but that just means that we can also just hire a Polish consultant and then send him out to the customers!?'— 'No, you can't, because he should be able to speak the language and he should be able to understand what business he is in.' And then I say: 'Okay, and developers do not have to do that!?' […] when there is an available resource we often talk about it in this way: 'but now they do not have much to do in one of the other departments, so can't we then borrow one from there?' —'well, yeah… you can do that, but it's typically difficult to get them here to complete a task. There can*

> *be certain types of tasks where you can say that it does not require you have domain knowledge' [...] but typically we just don't have a lot of such tasks, so."*
> ***(Senior Product Manager, 2nd Workshop)***

This senior product manager was essentially saying that there is a tendency to neglect the complexity of the work performed by a developer (in comparison with, say, the work of a consultant). Interestingly, when introducing implicit bias with the aim of addressing stereotypical descriptions of national boundaries and cross-cultural differences, different types of bias started to emerge, namely a bias that favors the consultants over the developers. This bias has a key impact on the way in which we consider and implement practices of substituting labor forces that occupy code programming in comparison to the work of the consultants. For instance, there is a current initiative in Enterprise IT that strives to unify and consolidate the various ways in which software is being developed in the different business units within Enterprise IT. The overall goal is to establish a common agile software development methodology framework. This initiative is motivated by a hope that software developers can—to a greater extent—be utilized and shared across different units, projects, and products. This is certainly not a trivial task and it has been highlighted that Enterprise IT has no fewer than 21 different software development units that implement different kinds of software development methods, use different kinds of software development tools, and work with various kinds of technology platforms, products, domains, cultures, and customers.

Raising attentiveness toward implicit bias within GSD organizations has the potential to mitigate the effects of employees' and managers' biases and prejudices. Discussing implicit bias in a legitimized way does not entail determining whether particular people are either good or bad; rather, it means acknowledging that we are all implicitly biased to some degree, and, as I demonstrated above, attending to implicit bias enabled collaborators to identify and discuss preconceptions that they had about global software outsourcing, which oftentimes were dominated by narratives on cross-cultural work challenges and national cultural behavioral patterns. By articulating and introducing implicit bias as an alternative lens and vocabulary to understand and discuss challenges of globally distributed collaboration, new or additional aspects of GSD began to emerge. In particular, I started to identify the presence of implicit bias at all managerial levels in the

organization: for example, the misconception about how developers are "just developers" who can (contrary to the consultants) be moved, substituted, or borrowed seamlessly within the different business units, divisions, or customer domains. This distinction between the developers and consultants represents a strong bias, which also supports some of the fundamental but slightly misinterpreted arguments for utilizing globally outsourced labor in software production and which I have previously criticized in an earlier study (Matthiesen and Bjørn, 2016), namely that software development work transcends geographical boundaries and thus can take place *anywhere* and *anytime* (Perry et al., 2001). This simplistic interpretation of globally distributed software work also becomes pertinent with the "Danish scrum meetings," which were then *translated* to the Indian software developers by pulling out 'only the necessary points' required for programming. Similarly I have pointed out how the lack of representation of remote workers at a physical Kanban board is problematic for maintaining transparency about the remote collaborators' work as well as building and improving relationships with the collaborators who are less engaged in the globally distributed collaboration (Matthiesen et al., 2014). The effects of such biases are that software development teams struggle in getting collaborative work to function, due to the lack of human, technological, or system support, as well as the lack of suitable team and task organization (Jensen and Nardi, 2014; Matthiesen et al., 2014; Bjørn et al., 2014b; Matthiesen and Bjørn, 2015; 2017). However, this is not just a reference to a simplistic interpretation of GSD work; it applies to the general simplistic view of the profession of the 'software developer' (Matthiesen et al., 2014) and the collaborative and intellectual work involved in software system development (Grinter, 2003; de Souza et al., 2004; Avram et al., 2009).

From the above, I learned that the importance of paying attention to and highlighting implicit bias is not just relevant in relation to GSD but indeed on a broader scale within a corporation. For example, I found how the induction of the company's 'own' (co-located) colleagues entailed more sympathetic treatment in relation to expectations and the contextual information and domain knowledge that a person needs to know in order to do the work. Thus, this research also points to the importance of providing structured and fair conditions for performing the work, to limit the risks of applying stereotypes that confirm existing but flawed prejudices.

This includes considerations of global/local task distribution and allocation, team distribution and allocation, team setup and rotation plans, (Hinds et al., 2015; Matthiesen and Bjørn, 2015; 2017), as well as closely coupled work and frequent communication (Bjørn et al., 2014b). My research indicates that paying attention to implicit bias as a new approach to address cultural stereotyping and unfold collaborative challenges in GSD should not only involve those who are directly involved in global work, but also those employees who make decisions on behalf of a GSD project; this includes the top-level management, project management, consultants, architects, and maybe even the customers.

8.2 Stereotyping Within the Collaborative Tools and System Structures

In the previous section, I discussed how various implicit biases are manifested in the ways in which globally distributed collaboration is put into practice; however, because work practices are highly related to the tools and systems serving to support the collaboration (Matthiesen and Bjørn, 2017), it is equally important to explore how implicit bias is manifested in the collaborative tools and system structures for GSD. In particular, it is vital to explore the underlying categories that make up a certain system or structure (Suchman, 1993), as these categories may support or reveal implicit preferences and biases that people have toward globally distributed collaboration.

One of the things I found particularly interesting when working in the GD department was the ways in which the general management of resources in Enterprise IT took place through a certain system for assigning employee initials. In short, these initials are unique identifiers consisting of five letters and representing a given employee. If, for example, your name is Kristina Larsen, then a typical initial could be 'KRILA,' which represents the first three letters in the given name and the first two letters in the surname. Initials serve as company usernames and as prefixes in staff email addresses (e.g., KRILA@enterpriseit.dk). The initials are often meaningfully compiled, like KRILA, and are useful for efficiently sending an email, looking up a colleague at *Skype for Business*, tracing an author of a document on *SharePoint*, and for managing employees in the ERP system. However, the logic that has—over time—been put into these initials is inopportune when we then look at the initials from a GD perspective. At some point in time it was decided that external

resources (external consultants, freelancers, and other temporary staff) would all receive initials starting with 'xx.' This serves as a handy way of rapidly sorting the employees, and in that way everyone can immediately identify any external resource just by glancing at the initials.

However, when it comes to the GD consultants, they are neither employed on the same contract terms as the Scandinavian Enterprise IT employees who are *internal permanent* staff, nor as *external temporary* consultants, but instead a significant portion of the global employees belong to an additional third category of *global and permanent* staff. However, one of the first things I discovered when I started out my fieldwork in Enterprise IT was the way the company assigns the GD consultants with same prefix initials (with two or three 'x's) as the temporary external consultants, which basically is an inaccurate categorization that leaves the names of the global collaborators obscure. While company usernames and initials may seem inconsequential, there are some issues related to this categorization scheme and a director who is managing a team in Poland reported that one of the biggest challenges he was dealing with was the onboarding of GD consultants due to current assigning of initials:

> *"System-wise we treat Warsaw colleagues as external consultants/subcontractors—this is not helping building a coherent team. Especially the 'xx' in their initials and mail addresses are an eyesore—people making the wrong conclusion."* ***(Director, January 2018)***

What is bothering the director is that Enterprise IT applies this crude way of sorting their employees, and according to him, the 'xx' carries some negative associations that impact people's willingness to cooperate, spend time with, or train a GD employee that is reckoned as 'external' within the available organizational and system structures. In particular, he explained to me how he had been working as an external consultant for many years and that, from his experience, this role of being an outsider has a huge impact on collaboration. Aside from the inherent values implied in the initials, these rather obscure initials were both difficult for collaborators to recognize, but more importantly they made very little sense when employees communicated directly with Enterprise IT's customers.

This discussion of IT-related policies regarding the organization of, and the structure for assigning, username initials and company email addresses may at first glance seem somewhat irrelevant to the task of improving tools for collaboration, as these username initials do not directly facilitate collaborative situations. Nevertheless, these initials proved to be significant for the ways in which people choose to interact or (dis)engage in collaborative work in the first place. Not only did a manager find the initials counter-productive toward the effort of establishing strong collegial relationships and collaboration across geographical sites, but also, for those business units facing customers directly, the initials were impersonal as they were hard to comprehend or even remember. According to several managers, the obscurity of the 'xx' initials mattered for how the developers working in GSD setups were perceived as either part of the company or as external, temporary staff. In this way, this structure somewhat signals that the GD consultants are kept at arm's length in the company and thus, I flagged my concerns with the GD VP and set out to address the problems with these initials. At first, I thought this would be a minor intervention and easily changed; however, as I will now explain, it proved to be a battle of greater dimensions than expected.

The system structures of sorting personnel in a company may be meaningful and useful in relation to legislation and terms of employment for a human resources department and thus, this structure may come across as a trivial or neutral organization of labor. However, such schemes for classifying and sorting (Bowker and Star, 1999) personnel may have critical consequences for collaboration, as specialists perceived to be temporary are often given menial tasks with low perceived value (Vora, 2015), both from the specialist's perspective as well as within the organization. It may also endorse unfeasible power relations within groups of the company employees and subgroups of the global employees (Durnell Cramton and Hinds, 2005; Hinds et al., 2015). Moreover, the onboarding procedures of the global employees proved to require additional manual work due to the insufficient classification scheme of staff, which the GD department were working on changing by further defining and incorporating categories for the GD consultants in the company infrastructures. This issue was further highlighted during the workshops, where discussion on implicit bias led to additional discussions about the systems and structures for supporting globally distributed collaboration. In particular, I

found it interesting how the workshops opened up to additional and sometimes exhaustive discussions of issues that the participants found most relevant to them. For instance, a manager said:

> *"I think we have discussed some relevant things in relation to what is happening in everyday life [...] I think there are many things we've talked about, which is relevant to talk about, just system-wise, it's never really been something that has been taken care of before, but merely something that has just become like: 'now we have a lot of external consultants' and then we're trying to make it work somehow"* ***(Resource Manager, 2nd Workshop)***

What the above quote demonstrates is that the systems or structures that are used in an organization are, for the most part, designed and implemented in a disconnected manner and rarely discussed, integrated, or adjusted afterwards. Instead, systems become standard and fade into the background (Bowker and Star, 1999), and additional structures and practices may even grow out of them as well. In the case with the username initials, particular system structures and schemes not only impacted the way labor forces were organized, managed, and perceived in the collaborative work with distributed colleagues, but also I found that they impacted the ways in which the work could be realized in practice within GD.

During a visit at the Warsaw office, I observed a situation where a few collaborators from Denmark and Poland were puzzled about some issues that they ran into during a test. In particular, a GD Consultant kept encountering an error when executing a certain test scenario that essentially was to create a folder and place a specific document within this folder. However, while the senior developer who was visiting from Denmark was able to carry out this scenario, the tester was only able to create a folder but not the document, and after some investigation the senior developer concluded that the reason was probably due to the differences in access constraints and author rights for temporary and external resources, which is automatically inferred by the 'xx' initials.

In my efforts to try and change the ways in which initials were assigned to the GD consultants, I attended various meetings with people from analytics and human resources. At these meetings, I discovered that changing the way initials were assigned to GD consultants was not as straightforward as first expected. Instead, I

discovered how initials were embedded into various economic, administrative, and analytical processes and practices in Enterprise IT and I encountered contradictory answers in relation to whether a change was possible. According to people in analytics, the initials could definitely be changed, but the process would require some changes in the way staff were configured in the Enterprise Resource Planning (ERP) system, and thus these changes entailed that the GD consultants would have to apply some of the same practices that the Scandinavian staff did in relation to time registration and performance measures. The GD department was already trying to apply these practices and so it seemed as the right way to move forward. And so, during several iterations and with the consent of the top management, the GD department developed new categories for the ways in which GD consultants should be administratively configured in Enterprise IT. However, just when I thought that the 'xx' prefix for GD consultants was about to become history, I met with resistance from the head of human resources who was not willing to discuss any changes on the basis of the current GD setup. Without going into too much detail, the controversy was due to the fact that these initials were utilized for categorization and sorting of staff in very different ways depending on the department. A key issue in this respect was the way personnel were analyzed in relation to performance and billable hours (actual hours a GD consultant can bill a customer). Another issue was the way personnel were contracted in relation to terms of employment, as this matters for how Enterprise IT is obliged to comply with the rules of employment law, including certain tax rules, insurance regulations, and the Employers' and Salaried Employees' Act.

Initially I thought that changing the ways initials were assigned in the company was easily fixed, which reveals my own implicit bias about the complexities of sorting staff in the human resources department. In the meantime, the GD department asked for permission and got clearance to create additional email addresses for the GD consultants so they could skip their 'xx'-related addresses and instead use email addresses based on their first names and surnames (i.e., *[firstname].[surname]@enterpriseit.dk*). Nevertheless, the username initials remained the same, and thus, issues in relation to access rights/constraints (such as the one from the test scenario) remained.

What the above example demonstrates is that, in order to address the use of cultural stereotyping and to really understand and mitigate challenges in collaborative work, there is a need to explore the relevant tools and systems as these may inscribe certain categories that foster or build upon various implicit biases among collaborators, which directly or indirectly impact the way people collaborate with each other. Moreover, different systems and structures within a corporation may serve different and incompatible purposes depending on the department that utilizes them. Thus, when exploring implicit bias and trying to uncover the challenges that these biases may bring about, it is not just a matter of identifying the systems that are inconvenient for global collaboration and articulating why these are problematic, but also it is a matter of figuring out how these systems can be altered in ways that keep things meaningful and take into account the different ways these structures or systems are applied in practice within the different departments. While I did not manage to change the way initials are assigned to GD consultants when I left the company, at least I contributed to make sure that people received email addresses with more meaningful prefixes that hopefully could help mitigate some of the implicit biases that existed around the 'xx' initials. Finally, I made sure to send out the message to the middle and top management at the annual outsourcing event in Enterprise IT where I presented my work and how I had tried to address negative stereotyping by bringing to attention the GSD Stereotype Framework and, in particular, implicit bias in Enterprise IT.

8.3 Stereotyping Within the Conditions for Work

In the previous section, I discussed how implicit bias may reside within the categories embedded into systems and structures that we apply to support collaborative work. Moreover, these structures and systems may also foster implicit bias among collaborators due to a lack of accurate feedback or support, or through the particular practices that have informally evolved along with using these systems. For the system for managing staff in Enterprise IT, implicit bias manifested itself in certain practices that were problematic for the global collaboration. I demonstrated that the particular ways in which staff were categorized mattered for what become the conditions for global work and how collaborative work could be accomplished. What we can take away from the example is that when technologies are designed, carefully selected, and put into practice in a particular context, the

technology defines a certain space in which it acts. In Suchman's terms, one can say that the technologies have agency (Suchman, 2007). The level of influence or *agency* that the technologies have, provide, or limit is not only based on the values and goals upon which the GSD arrangement is established, but it also reflects the things that are common sense to some and thus, taken for granted (Forsythe, 1999). The agency you voluntarily or involuntary inhabit—the position (physically, hierarchically) you are given—poses the conditions for performing the work in a given power hierarchy or professional relationship, and builds on the conscious or implicit presumptions that those involved have and the pre-history (the historical trajectories) of your location, gender, or ethnicity. Thus, it is important to explore how the conditions for work differ among people and across locations.

In the following, I further unfold how this categorization scheme for assigning usernames/initials affects and forms the particular conditions for work that a certain category of staff have or do not have, and how these conditions somewhat reflect the company's conscious or implicit view of the global outsourcing engagement. For instance, there are certain perks and company arrangements that only include permanent staff and thus, the question becomes: if you are a GD consultant and manager, are you then entitled to attend the annual management forum in the company and will you and your GD subordinates be invited to the company Christmas banquet and receive a Christmas present? These were some of the questions we discussed in the GD department as all GD consultants were categorized as external temporary staff even though the GD department found it important that the global staff would be included in the company on similar terms as the internal permanent staff. These discussions are indeed important for understanding how global work takes place in practice, but they are also important for identifying and understanding the strategic goals for engaging in global outsourcing. This leads to a discussion on what global offshore outsourcing entails in reality, and what the long-term perspectives are for these arrangements. Put differently: if companies really want to improve globally distributed collaboration, these relationships need to be unambiguous, clearly communicated, and adjusted accordingly within a company.

In Enterprise IT, one of the challenges that was frequently articulated by the people involved in GD concerned the lack of support that they received from the organization when trying to commence GD in practice. In particular, the GD department had ongoing discussions with people who found it difficult to operate within the space that the company offered for engaging in distributed collaboration. This was not a specific challenge for Enterprise IT but merely an indication of these discrepancies between what is expected from a global outsourcing engagement and how globally distributed collaboration is conditioned within a company, as I have also reported elsewhere (Matthiesen et al., 2014; Matthiesen and Bjørn, 2016; 2017). One of the things that people in Enterprise IT found problematic was a misalignment between the current incentive structures in the company, which included performance measures of, for example, *utilization.* Utilization is a measure of the number of actual hours that an employee spends on production (that is the hours that can be billed to the customer) in relation to the available hours (that is the total number of hours that the employee is being paid for by Enterprise IT). Despite the fact that measuring utilization is very common in the IT industry, there is also an agreement that this structure of a performance measure may promote disadvantageous incentives, some of which were discussed during one of the workshop sessions:

> ***GD VP:*** *"We are in this tight corner as we are all measured on utilization, for instance. So, if you spend a lot of time training Lukasz [a Polish GD consultant]..."*
>
> ***Managing Director:*** *"Yes, yes—then you're just being knocked on your head."*
>
> ***GD VP:*** *"Yes, so the question is: 'should we make sure that utilization goes up in Poland? Or should we just focus on securing our own utilization score, which is what* I *am measured on?'"*

Through the conversation above, we get an idea of how the performance measure in relation to utilization may perpetuate existing biases that are disadvantageous to establishing and maintaining collaborative relationships. According to one of the managers, another structure that may also hinder collaboration in Enterprise IT is the internal travel time:

> *"Well, I have heard that if you are to help a project in Sweden, for example, when you go to Sweden, it affects the travel time, so you have to travel during*

> *night time, otherwise you will be beaten on the head. It's just things like that, where there is inconsistent incitation"* ***(Managing Director, 1st Workshop)***

The structures discussed above not only apply to globally distributed work and GD, but these structures may also affect everyone who is being measured on, for example, utilization in the organization. However, something that is directly related to GD is the level of support provided by the organization and the top-level management:

> ***Senior Manager:*** *"Well, some of the iterations we have had in our department, we have looked at this and thought: 'Okay, if we have to go into this [using GD], it's an investment.' But we also know that this investment is an investment that we must undertake ourselves; there's no understanding [from the top management] if we jump out and take on this risk…"*
>
> ***Managing Director:*** *"… and are you also supposed to realize your business case in 2018?"*
>
> ***Senior Manager:*** *"Yes, yes, it's … like in February [laughing] yes, in Q1 [first quarter of the year], it should be home safe, right? At this point they [the stakeholders] should be able to see that we are on the right track, right? […] There is no understanding of jumping out and taking that risk [...] there is no understanding further up in the system, the Enterprise IT system, that this is the way we want to go! Enterprise IT have stated that we want some part [of the software production] to be produced through GD in 2018, but nothing else come in our way […] the risk involved in realizing this is something that each division must undertake themselves"*

The conversation above demonstrates how the managers find that there is a fundamental gap between the vision that Enterprise IT wants to undertake with GD and the ways in which risk-bearing and support are offered by the organization. This gap between the strategic initiative and the support structures was further confirmed in a questionnaire where "*Spending a large effort on a 'strategic initiative' that is not sufficiently supported by top-level management"* was stated as a potential weakness for Enterprise IT when utilizing GD. The issues presented here show how there is an underlying bias in the ways in which conditions for collaborative work are organized and supported. In particular, this bias concerns the favoring of the

individual's ability to perform well while ensuring as many work hours as possible are utilized and thereby billable to a customer. The utilization and performance measures affect the space in which the collaborative work can be established, maintained, and eventually grow into long-term formations that can offer strong business cases that the department dares to stake on. In particular, the existing performance measures risk promoting disadvantageous incentive structures as they affect the latitude of collaborative work, and thus, these utilization measures should not be neglected.

While there is no doubt that a significant responsibility for addressing the potential conflict zones between local and global software development lies with middle to upper management, there is also an opposite bias or a popular story present here. From the above example we learned that the utilization measures do not only apply to global work but also to the local work that takes place within the organization. This means that measurement of utilization is not a unique challenge for the global collaboration but for collaboration in general. Thus, a manager's struggle, hesitation, or lack of commitment toward incorporating GD in his or her business unit cannot solely be attributed to the upper management's lack of support for realizing the organization's strategic initiatives for GD. Instead, there is also the perspective of who is benefitting from putting the efforts into establishing globally distributed teams and work arrangements. Returning to one of the managers in the quote above, he talks about GD as 'investments,' which indicates that he is aware that it is also his department that will either benefit or suffer from the establishment of GD in his unit. While I do not attempt to choose sides here, what is demonstrated by the above example is the conflicting comprehensions of what are the challenges of getting GD work to function effectively. There are indeed structures that challenge the work and collaboration, which includes GD work; however, this may also be part of the normal, natural challenge for many collaborative arrangements within Enterprise IT. Either way, what is key to understand is that the particular conditions that are being provided by infrastructural aspects such as systems and structures need to also undergo investigation in terms of the ways in which these conditions are implicitly biased through, for instance, performance structures.

From the discussions at the workshops, it was evident how the organizational structures within the company impacted the ways in which people were encouraged or demotivated to engage in distributed collaboration. I learned how the various incentive structures and quantitative measures posed certain conditions for the kinds of collaborations that could take place across the various locations in the company. For example, a manager explained how there was a common view on how the travel time introduced with cross-local and global work has a negative impact on the ways in which the company measures people's performances, and thus, according to the manager, the consequence is that people feel obliged to perform a work-around, which includes night-time traveling. The quantifiable measures I have already discussed elsewhere (Matthiesen and Bjørn, 2017): measures of, and technological tool representations of, for example, the speed in which tasks were completed were found to take part in hiding the actual soundness and status of the distributed collaboration. From the workshop, it is likewise evident that the quantifiable measures matter for the software work and thus, for GSD work. For example, the measures of utilization—the number of actual billable hours that a developer spends on production—produces a disadvantage not only for the global work, but also for the general internal collaboration in the company as it turns out that individuals may be compelled to consider their own performance over the collective and collaborative team performance.

Organizational structures are indeed important to consider when exploring the conditions for engaging in global work. However, any account of conditions for work should also include a consideration of the physical and infrastructural aspects of working at a certain location, as I have demonstrated elsewhere (Matthiesen and Bjørn, 2016). One of the things I therefore prioritized when visiting the collaborators in Poland was exploring the concrete physical surroundings such as the office environments involved when working from Poland. This approach was further justified when I traveled to Poland with a new team and their managers to meet with their global collaborators and to kick-off a new project. We experienced being welcomed into the head office to meet and work during our stay. However, it turned out that this was in fact not the place from where the global collaborators performed their work on an everyday basis. There may be several reasons for why we were invited to the head office. For instance, it may be the case that the head office has

more room for hosting a full team, and it may have been deemed convenient that the office is located closer to the visitors' hotel. However, when a team and its stakeholders visit their IT vendor's site, it is important to have in mind that the team also represent the client/customer, which may also explain why we were invited into exemplary settings where great coffee was served and the fruit basket was full. Nevertheless, when I encouraged some of the collaborators from Denmark to go see the actual offices of the Polish collaborators, one of the things they realized was that the developers' desks were missing the additional monitors that were officially part of the contract deal and paid for on a monthly basis. For the manager, the missing monitors were a big deal, as he considered them to be crucial for the developers and testers to do proper work. This was, of course, a misunderstanding, and I relate this episode not to criticize the given IT vendor, but instead to demonstrate how minor but important physical settings all add to the distributed team's conditions for engaging in global work.

Exploring implicit bias in order to identify and potentially mitigate the collaboration-related challenges in global software development has certain potential while also posing different challenges. During the workshop and in my additional fieldwork, I saw several examples of how the Danish employees differentiated themselves from their remote colleagues by, for instance, explaining how a 'yes' has a different meaning depending on whether the person hearing it is a Dane or an Indian. Furthermore, I saw several examples of how the Danish employees differentiated their remote colleagues based upon their nationality. During the workshops, it was evident that participants displayed their explicit as well as implicit biases toward working with remote colleagues in different countries; for instance, the software developers working out of Poland were perceived differently than software developers working out of India. As I have presented above and elsewhere (Matthiesen and Bjørn, 2017), issues emerged due to the implicit biases about code programming as work highly structured through computerized tools and with little need for domain expertise and human interaction. This apparent lack of knowledge about the complexity of others' work (O'Neill et al., 2011) reinforces the argument for why software developers, unlike business consultants, can be distributed with little cost.

Similarly, I have earlier stressed how it is it is crucial to allow employees to articulate the problems they experience, since such insights provide important knowledge about the issues and concerns involved for those engaged in GSD (Matthiesen and Bjørn, 2017). In particular, the managerial tactics in GSD, whereby certain topics, criticisms, or experiences of the employees about global work are refuted and underestimated by management, risk leading to the failure of projects (Matthiesen and Bjørn, 2017). In structuring the workshop, the plan was to cultivate attentiveness towards negative stereotyping through implicit bias, and, together with the participants, to develop a new vocabulary that allowed participants to analyze and express their experiences more precisely and in a more nuanced manner, as a way to break with simplistic 'us and them' dichotomies. Just because people may not talk about the problems they experience, that does not mean the problems do not exist. In the organization studied here, it was evident that the workshop participants also experienced problems and the way the participants first articulated these problems was by using their existing vocabulary, and thus the participants would find themselves in an impasse whereby the current vocabulary focused primarily on the commonly agreed upon 'national cultural behaviors.' However, by attending to implicit bias as a new approach for combatting pervasive practices of deploying static cultural narratives and negative stereotypes in GSD, a space was created for the workshop participants to discuss their concrete challenges and experiences with GD. The implicit bias perspective made it possible to address the challenges that the participants experienced while also developing a new vocabulary about global work that rejected stereotypical language use.

Across all of my research, it is evident that organizations must take seriously the collaboration-related challenges that employees experience. These challenges I found stem from the lack of support and resources provided within the actual project (Matthiesen and Bjørn, 2017), from the lack of directions and guidance at a more general organizational level (Matthiesen et al., 2014), or from the organizational structures that encourage practices that are inconsistent with the task of creating incentives for establishing and maintaining well-functioning cross-local and distributed collaboration, as we saw in the case with Enterprise IT. However, I also found that the specifics of these challenges were rarely articulated as the primary issues, but instead these tended to be accounted for through

descriptions that built upon national cultural stereotypes. Thus, if we still want to move away from a shallow description and stereotypical understanding of global work, the task is not simply to listen to the complaints, some of which will be based upon somewhat clichéd and simplistic interpretations of a situation. Again, the workshop demonstrated that, to solve this issue, the provision of an alternative vocabulary is essential. In particular, by highlighting people's implicit biases, people are not only supported but also encouraged to move their discussions from the reporting of simplistic explanations about nationality towards an in-depth discussion on the underlying practices, organizational systems, and infrastructures that enable or constrain the global work.

In order for companies involved in GSD to move forward with developing their understanding of their implicit biases, the introduction of new theoretical explanations and an alternative vocabulary is critical. No single vocabulary exists to address all the challenges in GSD. While an awareness of implicit bias does pave the way toward articulating collaborative challenges in GSD in ways that seek to escape the prevalent interpretations of Hofstede's culture dimensions and the essentialist idea of national collective behavioral patterns (Søderberg and Holden, 2002; Walsham, 2002; Kwek, 2003), we still need additional research to develop a considerate and constructive vocabulary on collaborative challenges within GSD. It is not enough to simply have the vocabulary; we also need to embark upon a journey that includes additional research and development of activities so that new and nuanced perspectives can be introduced and embedded within GSD organizations. The GSD Stereotype Framework is one step toward addressing some of the important areas and attributes that deserve attention and investigation when trying to combat the use of negative stereotypes in GSD. In particular, my research centers on investigating and understanding the implicit biases that are expressed in the underlying classification schemes embedded in the practices, tools, and surroundings that construct the conditions for the collaborative work in GSD.

9 Conclusion

Since the founding of CSCW in the 1980s, there has been continued interest in exploring the nature of work in an effort to support the design of computer technologies for the workplace. It is unsurprising, then, that the distributed yet highly intertwined practices within the cooperative work arrangements in GSD are of particular interest for CSCW scholars (Bjørn et al., 2014a). In this dissertation, I set out to explore the coordination, communication, and use of cooperative technologies that global software developers engage in when involved in geographically distributed work within software offshore outsourcing setups in large organizations. Through 20 months of fieldwork between 2012 and 2018, I explored these complexities by conducting three in-depth and multisided ethnographic studies of GSD work and practices within three of the largest IT companies in Denmark and their global IT vendors.

Exploring this empirically I was—as an ethnographic CSCW researcher—committed to participating in social life and to encountering the world from the perspectives of the people involved in GSD work. I took this approach in order to further understand how people categorized their world and the specific vocabulary they used when describing their activities (Blomberg et al., 2003; Blomberg and Karasti, 2013). To investigate GSD work and practices from multiple perspectives, I traveled to different office sites in Denmark, Poland, and India. I investigated the various tensions that are at play when jobs and potentially power are relocated to foreign countries (Rost, 2004; Metiu, 2006; Jensen and Nardi, 2014). However, while my encounters with these tensions in the field were not always pleasant (as accounted for in Section 5.2), they made me more sensitive about my integrity and role as a researcher. Additionally, these experiences helped me direct my investigation toward the implicit issues and challenges that it seems companies hope will silently diminish as time goes by.

As I explained in Section 5.2, I first tried to keep culture in the periphery. Nevertheless, as I moved through the different empirical studies, I realized that, no matter how I tried to explore and unfold core CSCW concepts such as articulation work (Rönkkö et al., 2005; Boden et al., 2014), coordination (Cataldo et al., 2006;

Boden et al., 2007; Herbsleb, 2007; Avram et al., 2009; de Souza and Redmiles, 2009), common ground (Olson and Olson, 2000; Bradner and Mark, 2002), and closely coupled work (Jensen, 2014; Bjørn et al., 2014b), issues kept being explained through negative stereotypes describing how "Indian developers" or "Polish developers" think and behave. Therefore, I decided to allow myself to take note of and to further unfold how the explanations of distinct national cultural behaviors impacted the collaborative work in GSD. Nonetheless, as an ethnographic researcher, this change of course put me in a dilemma as it implied that I should apply the language that is used in the field and that is meaningful to those I studied (Blomberg and Karasti, 2013). The logic of my methodological approach, moreover, was that I should refrain from imposing any (or my own) outside meaning or theory on them (Brewer, 2000). Therefore, I tried my best to approach the field with an open mind and to use descriptive language to account for the all details that underlie the use of negative cultural stereotypes, which served as an explanatory fact for collaboration issues.

9.1 The Current Vocabulary in GSD and Why it is Problematic

Despite the fact that the stereotypical assumptions about how work is performed within a certain national culture have been rejected and criticized for a long time, the lack of alternatives is evident as we still witness 'magic bullet' thinking among the companies involved in cooperative work arrangements in GSD. As Kwek (Kwek, 2003) and Walsham (Walsham, 2002) have also pointed out: neither Hofstede's applicable theoretical framework of cultural dimensions, nor the various cultural crash courses—widely implemented within companies—offer sufficient details about how to deal with the real-life and situated issues in transnational work.

The four papers included in this dissertation (Matthiesen et al., 2014; Matthiesen and Bjørn, 2015; 2016; 2017) show that, underneath the "cultural differences," several considerations are often missing and these need to be taken into account when exploring collaborative challenges in GSD. For instance, the cultural blind spots demarcate what is focused on (or not) within an organization and among collaborators in a distributed arrangement. In the empirical case study discussed herein, the Danish collaborators had a blind spot regarding the involvement of the remote collaborators and the acknowledgement of their work efforts, and this became salient when digging further into the specifics of the way in which the daily

work was organized within the daily coordination and communication practices and supported through technologies (Matthiesen et al., 2014). Additionally, when simplistic descriptions (Ang, 2011)—about a particular group of distributed collaborators (such as those working from an office in India)—posit a perceived lack of effort or skill on the basis of a group or individual's ethnic or national background, we are obligated to further explore what underlies these simplistic descriptions if we want to fully understand the collaboration-related challenges. This includes exploring the available conditions for working and participating in the globally distributed cooperative work arrangement (Matthiesen and Bjørn, 2016). The conditions for work involve considerations on the locally and globally specific limitations and constraints for work, which, for instance, encompass system rights, hardware access, domain insights, knowledge support, work hours, or needs for daily commuting (Matthiesen and Bjørn, 2016; 2017). With these findings, this dissertation extends the conclusions of fellow researchers of GSD who have called for a more critical and holistic approach toward understanding how culture is intertwined with complex issues in GSD (Boden et al., 2009a), as they, too, find the cultural explanations to be a hindrance when it comes to solving collaboration-related challenges (Jensen and Nardi, 2014).

Both in theory and in practice, the culturally deterministic vocabulary, which serves as the underlying construct for describing and comprehending issues and challenges in GSD, contributes with insufficient accounts of the actual challenges for collaboration in GSD. The current cultural vocabulary contributes with simplistic interpretations of collaborative issues and challenges, failing to capture the underlying details and complexities that exist within the real-life and locally/globally situated collaborative practices in GSD. The simplistic explanations are problematic because they can cause significant rifts between people who believe they are very different from each other due to their national cultural heritages. Moreover, they offer a superficial symptom treatment that reinstates existing stereotypes and obviates any further or thorough analysis that has the ability to uncover the concrete, subtle, real-life, and situated challenges within the work and the collaborative relationships in GSD.

9.2 Moving Beyond Negative Stereotyping in GSD

To move beneath the surface of the use of negative stereotypes and to further the analytical conception of the coordination and communication challenges within the collaborative work and relationships in GSD, I asked: *How can researchers and IT professionals move beyond negative stereotyping and instead address the concrete coordination and communication practices that cause problems in global software development?*

In answering this research question, it was essential to first accept that there is a need for a broader vocabulary and a shift in focus. It was necessary to question the tendency of distributed collaborators to invoke 'culture' as a rhetorical strategy that closed down further discussions on collaboration in GSD (Jensen and Nardi, 2014). The task, therefore, was to figure out *where* the collaboration-related challenges materialize in GSD and *what* adds to the creation of stereotypes in GSD. While it would be naïve to think that the use of stereotypes in GSD can be removed completely, my interest was instead to describe the ways in which we can exploit the multiple co-existing stereotypes to identify the real-life and situated challenges of developing software within GSD arrangements. However, over time, once we develop a broader understanding and vocabulary that better captures the challenges in GSD, we can hope that the use of cultural rhetoric will appear as too narrow for justifying the collaborative issues we encounter in GSD.

Based on the findings from the four research papers included in this dissertation, I highlighted the various ways *categories*, *power hierarchies*, and *misconceptions* contribute to explaining the different collaboration-related challenges through stereotypical language constructs. For the categories—embedded in the global arrangement and instigative of certain practices—we know that they built upon conscious as well as unconscious and non-neutral assumptions about the world, society, culture, and domain wherein the classification scheme operates (Bowker and Star, 2000; Sengers et al., 2005). We also know that only by becoming more aware of our own unconscious assumptions (Sengers et al., 2005) are we are able to expand our purview and become more nuanced in, for example, our interpretation of collaborative work across various discontinuities. However, the question then becomes: how do we become more aware of the assumptions behind the particular

categories present in GSD? In answering this sub-question, I brought in *implicit bias* (Greenwald et al., 1998) as an alternative analytical approach that could help explain why even the most well-intentioned people draw on stereotypes and apply prejudiced descriptions of their foreign and remote colleagues when encountering issues in GSD.

The concept of *implicit bias* denotes the social stereotypes that we form outside our conscious awareness and that help confirm our existing (but sometimes incomplete) beliefs, while neglecting the complexity involved (Nickerson, 1998; Fiske, 2000). However, as I mentioned earlier, ethnography's epistemological stance differs significantly from that of psychology (where the concept of implicit bias is most commonly used), and thus it was necessary to explain how implicit bias is relevant for CSCW research. For me, implicit bias within the cooperative work arrangements in GSD is an analytical lens that enables us to put into words some of the subtleties that contribute to the application of stereotypes in GSD. This is valuable when investigating a domain like GSD where economic drivers, politics, and workplace realities converge into various real or assumed tensions that blur the picture of what are the concrete and situated challenges in GSD work. In this way, implicit bias as an analytical lens is complementary to the analysis of figuring out what initially adds to the production and use of negative stereotypical language in GSD. In particular, attending to implicit bias enables us to seek under the surface of stereotypical explanations while guiding us to refrain from entering discussions that condemn individuals as vicious or racist.

To bring to attention and identify the additional layers of complexities that risk leading to negative stereotyping, I developed a **GSD Stereotype Framework** that provides guidance for figuring out *where* collaborative challenges materialize upon narratives on "cultural differences." The GSD Stereotype Framework comprises three analytical ***areas***, which focus on 1) *the organization of work*; 2) *the collaborative technologies and system structures*; and 3) *the conditions for work*.

Regarding *the organization of work*, the practitioners and researchers need to analytically investigate the particular way the work is coordinated and communicated in practice and through tools (Matthiesen et al., 2014) or distributed

and supported in practice (Matthiesen and Bjørn, 2015; 2017). Here, one should attend to the ways in which the implicit biases and power hierarchies become manifested and thus, influence how the work is organized in GSD. This includes considerations on the ways in which scrum meetings are performed and documentation is produced in a particular local language that excludes foreign colleagues; or how the complexity of the work of developers is neglected in comparison to consultants' work due to the implicit biases and the inherent power hierarchies that exist among collaborators within the global work arrangement.

For *the collaborative technologies and system structures*, one should then attend to the ways in which the collaborative work is supported in practice, which includes considerations on *how* and *what* work is supported, and for *whom*. Central to explore are the categories embedded in the collaborative tools and systems, which have the ability to hide or marginalize important aspects of the work or the people in the collaboration (Star and Bowker, 2007). For instance, the categories in the Kanban board (Matthiesen et al., 2014), the sprint backlog, and burn down charts (Matthiesen and Bjørn, 2017) are important to investigate as they have the ability to hide actual issues or discrepancies in the distributed collaboration. Furthermore, I showed how the categories for sorting personnel had implications for the ways in which work could take place in practice due to certain system access rights or a lack of sufficient educational and onboarding structures. Additionally, power hierarchies and implicit biases are likewise attributes that deserve attention as these come to expression along with or through the categories that underlie the tools and system structures applied in GSD.

Finally, *the conditions for work* involve the infrastructural limitations or physical constraints for participating in the globally distributed collaboration. Here, the misconceptions that collaborators have about working at certain locations within the cooperative work arrangement in GSD are among the most prominent attributes we must direct our attention to. In particular, my research has demonstrated how the local infrastructural aspects and circumstances within GSD pose certain physical or structural conditions that are problematic for how, what, and when work can be accomplished (Matthiesen and Bjørn, 2015; 2016). This involves unequal, insufficient, or inappropriate travel/rotation policies, hardware access, daily

commutes, housing possibilities, and utilization and performance measures, which together form the available conditions for participating in GSD work at the different locations (Matthiesen and Bjørn, 2016; 2017). Nevertheless, the conditions for work that are offered at a certain location risk being unacknowledged or considered in an implicitly biased way, as these conditions may not always be salient to the collaborators working from a different office site. In this way, implicit bias enters as an important attribute when exploring the complexities of transnational collaboration, as distributed collaborators' work and work efforts risk being misinterpreted and explained through general descriptions that support or confirm existing biases, such as those that make gross distinctions between the intellects and skillsets of the Indian developers versus those of the Danish or Polish developers.

The GSD Stereotype Framework that I offer not only suggests where issues and challenges materialize in the first place and thus, where we should explore collaborative challenges in GSD. The framework also helps with identifying the attributes that contribute to the *creation, reinstatement*, or *use* of cultural stereotyping. Adding implicit bias to the GSD Stereotype Framework can remedy the application of various forms of stereotyping. This is because, when one's collaborative challenges are opened up and additional layers of complexity of the collaborative work in GSD are brought to one's attention, one can no longer be satisfied with applying stereotypical explanations to cover up the inappropriate organizations of work that are in place due to the particular power hierarchies and implicit biases involved in GSD. This also means that—if we are to move forward—neither can we disregard insufficient collaboration tools and system structures due to incomplete categorization schemes that favor, for example, work progression over quality. Nor can we ignore that misconceptions (for example, about the conditions of working from a certain location) become interpreted as outcomes of various stereotypical behaviors. The task, instead, is to ensure the right system or hardware access, or to accommodate utilization measures that motivate globally distributed collaboration. Thus, applying the GSD Stereotype Framework to our analysis ensures that explanations that are based on cultural differences and negative stereotypes can no longer adequately account for the complexity of the majority of the coordination and communication issues that are encountered in GSD.

9.3 Bringing the GSD Stereotype Framework to Practice

The GSD Stereotype Framework is a result of a bottom-up approach that relies on careful analysis of empirical data from the first two empirical studies. Furthermore, I expanded the analysis and thereby the attributes in the GSD Stereotype Framework by bringing in implicit bias as an analytical lens to further the conception of what creates or adds to the application of stereotypes in GSD. However, in order to provide answers to the research question, there remains the key question of what actions should be taken in practice to move beyond negative stereotyping and address the coordination and communication practices that risk causing problems in GSD.

In the third empirical study of this dissertation, I addressed negative stereotyping in practice. In particular, I was interested in bringing the GSD Stereotype Framework into practice and to further the conceptualization of implicit bias in GSD. I proactively delved into the subtle details of the collaboration-related issues and challenges that I discovered when collecting data and engaging in everyday conversations about global work in the company. I engaged in interventionist research approaches (Zuiderent-Jerak and Bruun Jensen, 2007; Bjørn and Boulus, 2011; Bjørn and Boulus-Rødje, 2015) as my mission was to bring actionable guidelines to the empirical field and bring to attention the categories, misconceptions, power hierarchies, and implicit biases that IT professionals need to reflect upon when involved in transnational collaboration.

I developed workshop and training material on distributed collaboration with a specific focus on moving beyond negative stereotyping by attending to implicit bias. Moreover, I presented preliminary findings from the two workshops at an internal global outsourcing event for the middle and upper management and the company's global IT vendors. The various attempts to intervene in the field and especially to bring forth implicit bias as an alternative gaze on globally distributed collaboration not only proved useful for discussing additional perspectives on how GSD work is enacted and supported within the company; it also created a space wherein people were allowed to air their issues and concerns that had previously been missing within an organization (Matthiesen and Bjørn, 2017). Bringing implicit bias to the field brought along a vocabulary that encouraged collaborators to stay attentive to the risk of applying national cultural stereotypes, thereby moving forward the

conversations about the challenges of collaboration. By using the GSD Stereotype Framework as a backbone, I was able to open up discussions about collaboration. This helped us to move us closer toward understanding the nature of the collaborative work and the locally specific and situated implications for supporting this kind of collaboration through technologies, which are at the core of CSCW (Schmidt and Bannon, 1992). In particular, the GSD Stereotype Framework can guide decisions about *where* the various interventions must take place in the collaborative work and *what* considerations need to be taken into account within organizations if they want to move beyond stereotyping and address the real-life and situated problems of coordination and communication in GSD.

Nevertheless, the framework does not make suggestions for *how* these interventions should take form. It is no easy task to intervene in the ways in which work is organized and practiced, collaborative technologies and systems are designed and used, and the conditions for work take form. I have demonstrated this point with the example of how username initials were assigned within a company, which revealed how a certain system structure is utilized in various ways across an organization. As the example showed, intervening effectively can also become a matter of figuring out how system alterations can be executed in a way that minimizes practical disruption to different departments.

Through this research, I extend related research on GSD by introducing the GSD Stereotype Framework, which offers concrete directions for *where* and *what* researchers and IT professionals (including those in managerial roles) need to attend to when wanting to identify in more detail the various collaboration-related challenges and issues in GSD. I argue that addressing negative stereotyping in GSD is a vital step toward unfolding the challenges of coordination and communication in more detail. In particular, I have shown how implicit bias can shed light on, explain, and further articulate certain power hierarchies (Hinds et al., 2015) that come along with the everyday work practices and efforts of reducing articulation work through, for example, various coordination practices in GSD (Boden et al., 2014; Matthiesen et al., 2014). Moreover, I extend research on categories (Winner, 1980; Suchman, 1993; Bowker and Star, 2000) by showing how implicit bias can help identify challenges that arise based upon the various categories that are not only

embedded in the collaborative tools, but that are also embedded or reflected in the local/global practices and conditions for accomplishing work within an organization. Moreover, I extend research that highlight the role of categorization as a device for sorting humans and their work efforts (Star and Strauss, 1999; Bowker and Star, 2000; Star and Bowker, 2007) by discovering challenges in relation to the implicitly biased and inadequate categorization of global permanent staff through username initials or the sorting of IT professions, which neglected the complexity of the IT developers' work over the consultants' work.

Exploring and seeking to address negative stereotypes in GSD is a way for a company to move forward if it wants to identify and figure out how to deal with collaboration-related issues in an effective manner. Additionally, it is worth mentioning that the username initials as well as the ways in which the work of certain professional roles are perceived differently demonstrate examples of implicit biases that are not related to ethnicity and national culture, but merely relate to the culture of the subgroups that belong to the particular organization, or a certain IT profession. In this way, my research echoes what fellow scholars have pointed out: that GSD work practices are not to be understood as radically different from the software development work practices that are not globally distributed (Avram et al., 2009). My research shows that attending to implicit bias is likewise relevant for comprehending collaborative issues and challenges within software development practices that are not distributed across nations. This means that, when unfolding the various coordination and communication issues that emerge in transnational settings, many of these issues could also emerge within distributed office sites located in the same country, which links back to the ways in which common ground is established (Olson and Olson, 2000) and collaborative work is enacted differently depending on the ways in which distance is perceived among collaborators (Bradner and Mark, 2002).

The contributions of this dissertation are not only relevant for understanding collaborative work in GSD. They apply, too—and perhaps to a higher degree—to large-scale software development in general. My work extends existing empirically informed literature on GSD as a collaborative practice (Herbsleb et al., 2000; de Souza et al., 2004; Boden et al., 2007; de Souza and Redmiles, 2008; Avram et al.,

2009; de Souza and Redmiles, 2009; Prikladnicki et al., 2013; Boden et al., 2014) within the area of corporate software offshore outsourcing, but hopefully it also adds valuable insights that can be utilized in common software development work as a discipline. To fully understand the complexity of (global) software development work and collaboration requires that both researchers and practitioners consider collaboration-related challenges from a perspective that moves beyond explaining collaboration issues through stereotypical descriptions of, for example, national, professional, or corporate cultural behavior. In this dissertation, I have demonstrated how attending to implicit bias serves as a timely and alternative strategy for IT and software development companies to apply if they want to move forward and excel in work and collaboration in the long run. Nevertheless, more research should focus more attention on figuring out—in more detail—how exactly companies should implement various interventions to raise attention to implicit bias as a strategic approach to address negative stereotyping and mitigate collaboration issues in the future.

10 References

Ang I. Navigating complexity: From cultural critique to cultural intelligence. Journal of Media & Cultural Studies. 2011 Nov 29;25(6):779–94.

Appadurai A. Modernity Al Large: Cultural Dimensions of Globalization. University of Minnesota Press; 1996.

Avram G. Knowledge Work Practices in Global Software Development. The Electronic Journal of Knowledge Management. 2007;5(4):347–56.

Avram G, Bannon L, Bowers J, Sheehan A, Sullivan D. Bridging, Patching and Keeping the Work Flowing: Defect Resolution in Distributed Software Development. Comput Supported Coop Work. Springer Netherlands; 2009 2009/12/01;18(5-6):477–507.

Bardzell S. Feminist HCI: taking stock and outlining an agenda for design. 2010.

Bertrand M, Mullainathan S. Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination. American Economic Review. 2004 Aug;94(4):991–1013.

Bjørn P, Bardram J, Avram G, Bannon L, Boden A, Redmiles D, et al. Global software development in a CSCW perspective. New York, New York, USA: ACM; 2014a. p. 301–4.

Bjørn P, Boulus N. Dissenting in reflective conversations: Critical components of doing action research. Action Research. 2011 Sep 1;9(3):282–302.

Bjørn P, Boulus-Rødje N. The Multiple Intersecting Sites of Design in CSCW Research. Comput Supported Coop Work. Springer Netherlands; 2015;24(4):319–51.

Bjørn P, Esbensen M, Jensen RE, Matthiesen S. Does Distance Still Matter? Revisiting the CSCW Fundamentals on Distributed Collaboration. ACM Trans. Comput.-Hum. Interact. ACM; 2014b Nov 21;21(5):27–26.

Bjørn P, Søderberg A-M, Krishna S. Translocality in Global Software Development: the Dark Side of Global Agile. Hum.-Comput. Interact. 2017 Dec 13;24(3):1–30.

Blomberg J, Burrell M, Guest G. An ethnographic approach to design. Julie AJ, Andrew S, editors. The human-computer interaction handbook. L. Erlbaum Associates Inc; 2003. p. 964–86.

Blomberg J, Giacomi J, Mosher A. Ethnographic field methods and their relation to design. … design: Principles and …. 1993.

Blomberg J, Karasti H. Reflections on 25 Years of Ethnography in CSCW. Comput Supported Coop Work. Springer Netherlands; 2013 2013/08/01;22(4-6):373–423.

Boden A, Avram G, Bannon L, Wulf V. Knowledge Management in Distributed Software Development Teams - Does Culture Matter? IEEE Computer Society; 2009a. p. 18–27.

Boden A, Nett B, Wulf V. Coordination Practices in Distributed Software Development of Small Enterprises. IEEE Computer Society; 2007. p. 235–46.

Boden A, Nett B, Wulf V. Trust and Social Capital: Revisiting an Offshoring Failure Story of a Small German Software Company. Wagner I, Tellioğlu H, Balka E, Simone C, Ciolfi L, editors. ECSCW 2009. Springer London; 2009b. p. 123–42.

Boden A, Rosswog F, Stevens G, Wulf V. Articulation spaces: bridging the gap between formal and informal coordination. Baltimore, Maryland, USA: ACM; 2014. p. 1120–30.

Bowker GC, Star SL. Sorting Things Out: Classification and Its Consequences. MIT Press; 1999.

Bowker GC, Star SL. Sorting Things Out. MIT Press; 2000.

Bradner E, Mark G. Why Distance Matters: Effects on Cooperation, Persuasion and Deception. New York, NY, USA: ACM; 2002. p. 226–35.

Brewer JD. Ethnography. Open University Press; 2000.

Carmel E, Argarwal R. The Maturation of Offshore Sourcing of IT Work. MIS Quarterly. Executive. 2002;:65–77.

Castro LA, González VM. Transnational Imagination and Social Practices: A Transnational Website in a Migrant Community. Human–Computer Interaction. 2014.

Cataldo M, Wagstrom PA, Herbsleb JD, Carley KM. Identification of Coordination Requirements: Implications for the Design of Collaboration and Awareness Tools. New York, NY, USA: ACM; 2006. p. 353–62.

Clark HH, Brennan SE. Grounding in Communication. American Psychological Association. 1991;:127–49.

Cooper A. The inmates are running the asylum. Indianapolis: SAMS; 1999.

Dabbish L, Stuart C, Tsay J, Herbsleb J. Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. New

York, NY, USA: ACM; 2012. p. 1277–86.

Daniel S, Agarwal R, Stewart KJ. The Effects of Diversity in Global, Distributed Collectives: A Study of Open Source Project Success. Information Systems Research. 2013;24(2):312–33.

de Souza CRB, Redmiles D, Cheng L-T, Millen D, Patterson J. How a good software practice thwarts collaboration: the multiple roles of APIs in software development. Newport Beach, CA, USA: ACM; 2004. p. 221–30.

de Souza CRB, Redmiles DF. An empirical study of software developers' management of dependencies and changes. Leipzig, Germany: ACM; 2008. p. 241–50.

de Souza CRB, Redmiles DF. On The Roles of APIs in the Coordination of Collaborative Software Development. Comput Supported Coop Work. 2009 Sep 16;18(5-6):445–75.

Durnell Cramton C, Hinds PJ. Subgroup Dynamics in Internationally Distributed Teams: Ethnocentrism or Cross-National Learning? Research in Organizational Behavior. 2005;26(0):231–63.

D'Mello M, Sahay S. "I am kind of a nomad where I have to go places and places"… Understanding mobility, place and identity in global software work from India. Information and Organization. 2007 Jan;17(3):162–92.

Eisenhardt KM. Building Theories from Case Study Research. The Academy of Management Review. 1989;14(4):532–50.

Esbensen M, Bjørn P. Routine and Standardization in Global Software Development. In Proceedings of the 18th International Conference on Supporting Group Work (GROUP '14). ACM, New York, NY, USA, 12-23.

Esbensen M, Tell P, Cholewa JB, Pedersen MK, Bardram J. The dBoard: A Digital Scrum Board for Distributed Software Development. New York, NY, USA: ACM; 2015. p. 161–70.

Espinosa JA, Slaughter SA, Kraut RE, Herbsleb JD. Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development. Organization Science. 2007;18(4):613–30.

Faraj S, Sproull L. Coordinating Expertise in Software Development Teams. Manage. Sci. 2000;46(12):1554–68.

Fiske ST. Stereotyping, prejudice, and discrimination at the seam between the centuries: evolution, culture, mind, and brain. European Journal of Social Psychology. John Wiley & Sons, Ltd; 2000 May 1;30(3):299–322.

Forsythe DE. "It's Just a Matter of Common Sense": Ethnography as Invisible

Work. Comput Supported Coop Work. Kluwer Academic Publishers; 1999 1999/03/01;8(1-2):127–45.

Fugelli P, Lahn LC, Mørch AI. Shared prolepsis and intersubjectivity in open source development: expansive grounding in distributed work. San Antonio, Texas, USA: ACM; 2013. p. 129–44.

Gerson EM. Reach, Bracket, and the Limits of Rationalized Coordination: Some Challenges for CSCW. Resources, Co-Evolution and Artifacts. Springer London; 2008. p. 193–220.

Gerson EM, Star SL. Analyzing Due Process in the Workplace. ACM Trans. Inf. Syst. New York, NY, USA: ACM; 1986 Jul;4(3):257–70.

Greenwald AG, McGhee DE, Schwartz JLK. Measuring individual differences in implicit cognition: The Implicit Association Test. Journal of Personality and Social Psychology. 1998;74(6):1464–80.

Grinter RE. Recomposition: Coordinating a Web of Software Dependencies. Comput Supported Coop Work. Kluwer Academic Publishers; 2003 Sep 1;12(3):297–327.

Halverson CA, Ellis JB, Danis C, Kellogg WA. Designing Task Visualizations to Support the Coordination of Work in Software Development. New York, NY, USA: ACM; 2006. p. 39–48.

Herbsleb JD. Global Software Engineering: The Future of Socio-technical Coordination. IEEE Computer Society; 2007. p. 188–98.

Herbsleb JD, Mockus A, Finholt TA, Grinter RE. Distance, dependencies, and delay in a global collaboration. Philadelphia, Pennsylvania, USA: ACM; 2000. p. 319–28.

Hinds P, McGrath C. Structures that work: social structure, work structure and coordination ease in geographically distributed teams. Banff, Alberta, Canada: ACM; 2006. p. 343–52.

Hinds P, Retelny D, Cramton C. In the Flow, Being Heard, and Having Opportunities. CSCW '15. New York, New York, USA: ACM Press; 2015. p. 864–75.

Hjalmarsson A, Gustafsson E, Cronholm S. Exploring the Use of Personas in User-Centered Design of Web-based e-services. iConference. iSchools; 2015.

Hofstede G. Culture's consequences: International differences in work-related values. 1984.

Hofstede G. Cultures and Organizations: Software of the Mind. London/New

York: McGraw Hill; 2010.

Hossain E, Bannerman PL, Jeffery DR. Scrum practices in global software development: a research framework. Torre Canne, Italy: Springer-Verlag; 2011. p. 88–102.

Howcroft D, Trauth EM. The implications of a critical agenda in gender and IS research. Information Systems Journal. Blackwell Publishing Ltd; 2008;18(2):185–202.

Irani L, Vertesi J, Dourish P, Philip K, Grinter RE. Postcolonial computing: a lens on design and development. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 1311-1320.

Jensen RE. Why closely coupled work matters in global software development. Sanible Island, Florida, USA: ACM; 2014.

Jensen RE, Bjørn P. Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds. Dugdale J, Masclet C, Grasso MA, Boujut J-F, Hassanaly P, editors. From Research to Practice in the Design of Cooperative Systems: Results and Open Challenges. Springer London; 2012. p. 123–36.

Jensen RE, Nardi B. The Rhetoric of Culture as an Act of Closure in a Cross-national Software Development Department. Tel Aviv, Israel; 2014.

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

Krishna S, Sahay S, Walsham G. Managing cross-cultural issues in global software outsourcing. Communications of the ACM. 2004;47(4):62–6.

Kwek D. Decolonizing and Re-Presenting Culture's Consequences: A Postcolonial Critique of Cross-Cultural Studies in Management. Postcolonial Theory and Organizational Analysis. 2003.

LeDoux J. The emotional brain: The mysterious underpinnings of emotional life. New York: Simon and Schuster; 1996.

Luff P, Hindmarsh J, Heath C. Workplace Studies. Cambridge University Press; 2000.

Marlow J, Dabbish L. Activity traces and signals in software developer recruitment and hiring. San Antonio, Texas, USA; 2013. p. 145–56.

Marlow J, Dabbish L, Herbsleb J. Impression formation in online peer production: activity traces and personal profiles in github. San Antonio, Texas, USA; 2013. p. 117–28.

Marsden N, Haag M. Stereotypes and Politics: Reflections on Personas. New York, NY, USA: ACM; 2016. p. 4017–31.

Matthiesen S, Bjørn P. Why replacing legacy systems is so hard in global software development: An information infrastructure perspective. New York, New York, USA: In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15). ACM, New York, NY, USA, 876-890.

Matthiesen S, Bjørn P. Let's Look Outside the Office: Analytical Lens Unpacking Collaborative Relationships in Global Work. COOP 2016: Proceedings of the 12th International Conference on the Design of Cooperative Systems, 23-27 May 2016, Trento, Italy. Cham: Springer International Publishing; 2016. p. 107–22.

Matthiesen S, Bjørn P. When Distribution of Tasks and Skills Are Fundamentally Problematic: A Failure Story from Global Software Outsourcing. PACM on Human-Computer Interaction. 2017 Nov;1(CSCW).

Matthiesen S, Bjørn P, Petersen LM. "Figure out how to code with the hands of others": recognizing cultural blind spots in global software development. In Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing (CSCW '14). ACM, New York, NY, USA, 1107-1119.

Mesman J. Disturbing Observations as a Basis for Collaborative Research. Science as Culture. Routledge; 2007;16(3):281–95.

Metiu A. Owning the Code: Status Closure in Distributed Groups. Organization Science. 2006 Aug;17(4):418–35.

Moss-Racusin CA, Dovidio JF, Brescoll VL, Graham MJ, Handelsman J. Science faculty's subtle gender biases favor male students. Proceedings of the National Academy of Sciences. National Academy of Sciences; 2012 Oct;109(41):16474–9.

Muller M. Feminism asks the "Who" questions in HCI. Interacting with Computers. Elsevier Science Inc; 2011 Sep;23(5):447–9.

Neeley TB, Hinds PJ, Cramton CD. The (Un)Hidden Turmoil of Language in Global Collaboration. Organizational Dynamics. Elsevier Inc; 2012 Jul 5;41(3):236–44.

Neyland D. Organizational Ethnography. SAGE; 2007.

Nickerson RS. Confirmation bias: A ubiquitous phenomenon in many guises. Review of General Psychology. 1998;2(2):175–220.

Nielsen L, Nielsen KS, Stage J, Billestrup J. Going Global with Personas.

Human-Computer Interaction – INTERACT 2013. Berlin, Heidelberg: Springer, Berlin, Heidelberg; 2013. p. 350–7.

Nielsen L, Storgaard Hansen K. Personas is Applicable: A Study on the Use of Personas in Denmark. New York, NY, USA: ACM; 2014. p. 1665–74.

O'Neill J, Martin D, Colombino T, Grasso A. When a little knowledge isn't a dangerous thing. CHI '11. New York, New York, USA: ACM; 2011. p. 1667–76.

Olson GM, Olson JS. Distance matters. Hum.-Comput. Interact. 2000;15(2):139–78.

Orlikowski WJ. Categories: Concept, content, and context. Comput Supported Coop Work. Kluwer Academic Publishers; 1994 Mar;3(1):73–8.

Paradies Y, Priest N, Ben J, Truong M, Gupta A, Pieterse A, et al. Racism as a determinant of health: a protocol for conducting a systematic review and meta-analysis. Systematic Reviews. BioMed Central; 2013 Sep;2(1):85.

Parnas DL. On the criteria to be used in decomposing systems into modules. Communications of the ACM. ACM; 1972;15(12):1053–8.

Perry M, O'hara K, Sellen A, Brown B, Harper R. Dealing with Mobility: Understanding Access Anytime, Anywhere. ACM Trans. Comput.-Hum. Interact. New York, NY, USA: ACM; 2001 Dec;8(4):323–47.

Prikladnicki R, Boden A, Avram G, Souza CB, Wulf V. Data collection in global software engineering research: learning from past experience. Empir Software Eng. Springer US; 2013 2013/01/16;:1–35.

Randall D, Harper R, Rouncefield M. Fieldwork for Design: Theory and Practice. Springer; 2010.

Rode JA. A theoretical agenda for feminist HCI. Interact. Comput. British Informatics Society Limited; 2011 Sep 1;23(5):393–400.

Rost J. Political Reasons for Failed Software Projects. IEEE Software. Los Alamitos, CA, USA; 2004 Nov;21(6):104–103.

Rönkkö K, Dittrich Y, Randall D. When Plans do not Work Out: How Plans are Used in Software Development Projects. Comput Supported Coop Work. 2005;14(5):433–68.

Schmidt K, Bannon L. Taking CSCW seriously. Comput Supported Coop Work. Kluwer Academic Publishers; 1992 1992/03/01;1(1-2):7–40.

Schultze U. A Confessional Account of an Ethnography about Knowledge Work. MIS Quarterly. Management Information Systems Research Center,

University of Minnesota; 2000;24(1):3–41.

Sengers P, Boehner K, David S, Kaye J'. Reflective Design. New York, NY, USA: ACM; 2005. p. 49–58.

Sharp H, Dittrich Y, de Souza CRB. The Role of Ethnographic Studies in Empirical Software Engineering. IIEEE Trans. Software Eng. 42(8):786–804.

Shklovski I, Lindtner S, Vertesi J, Dourish P. Transnational times: locality, globality and mobility in technology design and use. Copenhagen, Denmark: ACM; 2010. p. 515–8.

Shklovski I, Vertesi J, Lindtner S. Introduction to this Special Issue on transnational HCI. Human–Computer Interaction. 2014.

Star SL, Bowker GC. Enacting silence: Residual categories as a challenge for ethics, information systems, and communication. Ethics and Information Technology. Springer Netherlands; 2007 Sep 6;9(4):273–80.

Star SL, Strauss A. Layers of Silence, Arenas of Voice: The Ecology of Visible and Invisible Work. Comput Supported Coop Work. Kluwer Academic Publishers; 1999 Mar;8(1-2):9–30.

Stone J, Moskowitz GB. Non-conscious bias in medical decision making: what can be done to reduce it? Medical Education. Blackwell Publishing Ltd; 2011;45(8):768–76.

Strauss A. THE ARTICULATION OF PROJECT WORK: AN ORGANIZATIONAL PROCESS. Sociological Quarterly. Blackwell Publishing Ltd; 1988a;29(2):163–78.

Strauss A. The articulation of project work: An organizational process. The Sociological Quarterly. 1988b;29(2):163–78.

Suchman L. Do categories have politics? Comput Supported Coop Work. 1993;2(3):177–90.

Suchman L. Making work visible. Communications of the ACM. 1995.

Suchman L. Human-Machine Reconfigurations: Plans and Situated Actions. Cambridge University Press; 2007.

Søderberg A-M, Holden N. Rethinking Cross Cultural Management in a Globalizing Business World. International Journal of Cross Cultural Management. 2002 Apr 1;2(1):103–21.

Søderberg A-M, Krishna S, Bjørn P. Global Software Development: Commitment, Trust and Cultural Sensitivity in Strategic Partnerships. Journal of International Management. 2013 Dec;19(4):347–61.

Šmite D, Britto R, van Solingen R. Calculating the Extra Costs and the Bottom-line Hourly Cost of Offshoring. Piscataway, NJ, USA: IEEE Press; 2017. p. 96–105.

Šmite D, van Solingen R. What's the True Hourly Cost of Offshoring? IEEE Softw. 2016;33(5):60–70.

Uhlmann EL, Cohen GL. Constructed Criteria: Redefining Merit to Justify Discrimination. Psychological Science. 2005;16(6):474–80.

Vlaar PWL, van Fenema PC, Tiwari V. Cocreating understanding and value in distributed work: how members of onsite and offshore vendor teams give, make, demand, and break sense. MIS Q. Society for Information Management and The Management Information Systems Research Center; 2008 Jun 1;32(2):227–55.

Vora K. Life Support: Biocapital and the New History of Outsourced Labor. University of Minnesota Press; 2015.

Walsham G. Interpretive case studies in IS research: nature and method. European Journal of Information Systems. 1995;4(74):74–81.

Walsham G. Cross-cultural software production and use: a structurational analysis. MIS Q. 2002;26(4):359–80.

Walsham G. ICTs and Global Working in a Non-Flat World. Barrett M, Davidson E, Middleton C, DeGross JI, editors. Information Technology in the Service Economy: Challenges and Possibilities for the 21st Century. Boston, MA: Springer US; 2008. p. 13–25.

Watson-Manheim MB, Chudoba KM, Crowston K. Perceived discontinuities and constructed continuities in virtual work. Information Systems Journal. Blackwell Publishing Ltd; 2012;22(1):29–52.

Winner L. Do Artifacts Have Politics? Daedalus. The MIT Press; 1980;109(1):121–36.

Winograd T, Flores F. Understanding Computers and Cognition: A New Foundation for Design. Addison-Wesley; 1987.

Zimmermann M. Neurophysiology of Sensory Systems. Fundamentals of Sensory Physiology [Internet]. Berlin, Heidelberg: Springer, Berlin, Heidelberg; 1986. p. 68–116. Retrieved from: https://doi.org/10.1007/978-3-642-82598-9_3

Zuiderent-Jerak T, Bruun Jensen C. Editorial Introduction: Unpacking “Intervention” in Science and Technology Studies. Science as Culture. 2007 Sep;16(3):227–35.

PART II

Paper 1

Awarded Honorable Mention
Matthiesen S, Bjørn P, Petersen LM. “Figure out how to code with the hands of others”: recognizing cultural blind spots in global software development.
In Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing (CSCW '14). ACM, New York, NY, USA, 1107-1119.

"Figure Out How to Code with the Hands of Others": Recognizing Cultural Blind Spots in Global Software Development

Stina Matthiesen[1] **Pernille Bjørn**[1, 2] **Lise Møller Petersen**[1]

[1]The IT University of Copenhagen
Rued Langgaards Vej 7, 2300 Copenhagen,
Denmark

[2]University of California, Irvine
Irvine, CA 92697
USA

{matthiesen, pbra, lmp}@itu.dk

ABSTRACT

We report on an ethnographic study of an outsourcing global software development (GSD) setup between an Indian IT vendor and an IT development division of a Danish bank. We investigate how the local IT development work is shaped by the global setup in GSD and argue that the bank had cultural blind spots toward the changes in Denmark. Three critical issues were neglected due to the cultural blind spots: 1) increased number of interruptions, 2) lack of translucence of remote colleagues' work, and 3) the re-definition of boundaries between work and articulation work. The implications of these findings include considerations for how to organize GSD practices and prepare the organizational changes that occur when moving from a co-located software development organization to an inter-organizational geographically distributed organization. Also, our findings open up discussions about the professional identity of IT developers within GSD, including extending the qualifications for IT developers.

Author Keywords

Global software development (GSD); ethnographic study; cultural blind spots; articulation work; local work practices

ACM Classification Keywords

K.4.3 [Organizational Impacts]: Computer-supported collaborative work; H.5.3 [Group and Organization Interfaces]: Computer-supported cooperative work

INTRODUCTION

The CSCW community has an increased interest in global work practices within software development. Working remotely across time zones and geography impact the local work practices of IT developers, and issues such as trust [e.g. 1], coordination [e.g. 12], and awareness [e.g. 14] are crucial enablers for global software development (GSD). For companies determined to reap the benefits of global software development, methods for assuring the quality of software engineering processes and end product have tended to steal the focus, while the actual concrete practices have attracted less attention. Echoing the call for further understanding of actual concrete practices in software development through "*empirical studies of software development teams using extensive field study methods*" [3, p. 480], we present a study that focuses on local practices in a global collaborative setup.

In this paper we ask: *How is the local work shaped by the global setup in GSD?* Answering this question, we provide empirical data from an outsourcing setup between a large Indian IT vendor and a Danish financial enterprise that have been collaborating for the last 7 years. Interestingly, in this case the local Danish developers still articulate and experience many problems and challenges. At first glance, the obvious explanation for this observation might entail linking the complexities of working cross-culturally to "*Hofstede-type studies*" that understand cultures as nationally determined characteristics. However, this understanding poorly links to detailed work-related attitudes and actions when used as a basis for analyzing cross-cultural working [32]. Thus, if we are to move away from this "stereotypical" understanding and instead dive into the empirical complexities, a totally different set of reasons emerges.

We use the notion of "cultural blind spots" to encompass an unidentified knowledge that exists in parallel within organizations [9]. Cultural blind spots can be understood as unnoticed tacit knowledge we have about ourselves in relation to our culture, mind, action, or motivation that are typically overlooked when paying attention elsewhere. Cultural blind spots may encumber the collaboration, as people and organizations may have blind spots toward their cultural self-awareness. In relation to knowledge sharing in organizations and the processes through which people manage to act appropriately within a certain context, the cultural blind spots then represent unattended knowledge that, if attended to, may be a resource in the organizational context.

CSCW'14, February 15 - 19 2014, Baltimore, MD, USA

http://dx.doi.org/10.1145/2531602.2531612

We propose that one of the major reasons for the challenges in our case concerns the cultural blind spots of the Danish IT division toward the Danish local site. Although the Danish IT division has focused its attention toward the Indian IT vendor, ensuring alignment of processes and practices toward the Danish bank, it has paid less attention to the changes at the local Danish site. In this paper, we point to how this unbalanced attention from Danish IT division has brought about issues and concerns from the Danish site. Moreover, we point to how three crucial cultural blind spots in particular have been neglected by the Danish IT division: 1) how the global setup brings with it an increased number of interruptions that forces the coordinating and prioritizing of tasks to be reviewed differently at the local Danish site, 2) how the local work related to coordination within projects needs realignment to include translucence of the global collaboration and the work of the remote colleagues, and 3) how boundaries between work and articulation work at the local Danish site are re-defined when engaging in GSD.

The paper begins with a presentation of related work followed by a description of the research method. Then, approaching an answer to our research question, the case is described and the empirical results are presented. Finally, findings are discussed before the paper concludes.

RELATED WORK

Global software development is characterized by collaboration between IT developers with different national and organizational cultures in different geographic locations and across time zones using various traditional and IT-enabled means to collaborate [15, p. 88]. GSD is widely studied within CSCW, both in terms of open-source development [14, 22] and in corporate settings [6, 8, 17, 30]; our focus is on the latter.

Working across organizational boundaries entails malleable structures around the collaborative setup [21], providing opportunities to continuously align and readjust the global work arrangement. It is important to remember that people work locally and are part of the local organizational setup even when working globally, since the global organizational context is a conglomeration of the local organizational setup people bring to the table [5]. Thus, the local organizational setup impacts the global setup. In addition, the global setup impacts local work of the IT developers, since the developers also have to readjust and align to the new ways of working.

The question then becomes, which practices in the local organizational context are impacted by the global setup? Looking into the literature, the key collaborative practices in GSD are coordination [12, 14], knowledge management [2, 6, 17], organizational culture [6, 32], and trust/commitment [1, 30]. Examining how the local collaborative practices become impacted by the global setup, it is essential to realize that GSD is not the opposite of local software development [3] but rather software development with additional complexities in handling the extra effort of articulation work, which must be present to reach the global objective. Our interest is to "localize the global" [19, p.173 ff] in GSD, which means that we will study the global engagement through the local practices.

Articulation work is the extra work required to handle the mutual dependencies when more than one person is involved in solving a task [31]. Articulation work as a theoretical concept aims to help us understand the interdependence of activities where more than one individual is needed to solve a task. Articulation work has been studied in the practical setting of global software development, where it was found to be essential for handling the large number of interdependencies between the developers [7]. The complexity of articulation work in the context of GSD tends to be overlooked by the IT developers, even though it plays an important role in the management of distributed projects [7]. It is important to note that work for some professions might be articulation work for others. For example, a secretary might have it as her work task to organize meetings, whereas for a system developer organizing meetings would be articulation work. Therefore, what counts as work for a particular profession influences what counts as articulation work for the same profession. In this way, activities might be identified as articulation work, depending on the role of the employee.

Articulation work in GSD is closely related to coordination. Working remotely while being mutually dependent on each other for completion of interdependent work requires a lot of communication to mediate and control the cooperative relationship. Distribution of tasks across organizational, spatial, and temporal boundaries has been referred to as increased reach [12]. Increased reach has consequences for the organization of work and "*implies that the problem of coordinating tasks and people (never easy) becomes markedly more complex*" [12, p. 195].

Agile methods, originally assumed for collocation, are increasingly applied in GSD in an effort to mitigate the challenges of coordination stipulated by temporal, geographical, and socio-cultural distance [15]. Still, issues of increased coordination costs and inconsistent work practices impact coordination, and several papers [e.g. 6, 14] report reduced cooperation arising from misunderstandings. When we investigate how the local work is shaped by the global setup, we will include studying the local coordination practices as an approach to comprehend the global coordination practices.

Misunderstandings are closely related to how knowledge is managed between geographically dispersed participants [2]. Being able to communicate as well as to interpret the requirements of an IT system [8] to achieve a shared understanding of the work [14] poses great challenges. When operating in geographically distributed and intercultural collaborative work settings, distinct

interpretations of concepts and meanings were found to challenge the communication of shared tasks [17]. Based on the experiences that we collect as individuals and as part of groups, we are each left with an unarticulated frame of reference that we act on and live in [5]. When people's frames of reference are distinctly different, a shared meaning may be difficult to reach. Acknowledging the necessity of knowledge and skill in a team, it is not sufficient if team members cannot coordinate their expertise [11]. Technologies such as instant messaging offer synchronous communication. However, when answers are not needed instantly, an asynchronous use has been reported: sending off a question and continuing to work until the answer arrives later [7]. Instant messaging has been found useful in supporting knowledge work in GSD [2]. But, interestingly, the tools only played a secondary role; instead, the human factors such as communication and social skills were critical for success. The exchange of knowledge is a challenge further increased in situations where domain and task experts are distributed [13]. For example, if the majority of the domain experts are located at the headquarters in Germany, less motivation for knowledge exchange was found at the headquarters in comparison to the subsidiaries in, for example, Italy and India [13]. Here the sub-grouping of experts is overlapping with the sub-grouping of locations, thus increasing the impact of the sub-groups on the collaboration [10], making the us/them division more pertinent. The distribution and strict division of labor has also been reported to cause problems when dealing with exceptional cases in work processes where workers are lacking understanding and knowledge of the work of other division [25]. Examining the empirical data in our case, we will study sub-group dynamics as well as knowledge practices to understand how the global setup impacts the local work of the IT developers.

The socio-emotional relations between colleagues at the local sites are critical for the collaboration and are based on the trust and expectations toward remote colleagues [1, 30]. Social relationships are related to cultural aspects of the local contexts. Interestingly, asymmetry in cultural training in inter-organizational GSD setups has been reported [18]. In these cases it was perceived as necessary for the staff from the IT vendor to learn about the culture of their client organization's country and not the other way around. In these cases culture is perceived as a stable national entity that can be "translated" for others to understand. Studying cultural complexities, we join others by examining culture not as a stable entity based on nationality, but rather as a dynamic entity which gets re-negotiated and which we can only study through its manifestation in artifacts, practices, and routines [6, 17]. Understanding culture in GSD thus requires us to study the local practices and the negotiated processes of language, practices, and artifacts. Often cultural differences emerge in situations of conflict or communication breakdown [5], since these are the situations where differences become pertinent in practice. However, culture also manifests in local practice, even though these practices often appear as blind spots locally. The term *blind spot* comes from human biology and refers to a small region of the eye where there are no cells to detect light, and thus a part of the visual field is obscured. The brain typically compensates for the blind spot based on the surrounding details and information from the other eye. Which visual information falls into the blind spot differs depending on where the eye is directing attention. When attention is moved, the blind spot moves as well; however, the human eye can never access the visual information that falls in the blind spot directly. Cultural blind spots comprise the knowledge that exists in parallel within organizations without being identified [9]. For example, organizations might focus on creating a "global mindset" while completely ignoring the many international experiences that already exist at the organization's headquarters. So, by focusing on "something," other important knowledge might become opaque and out of sight. In many cases information that falls into the organizational cultural blind spot is about our own cultural behaviors and practices, since we simply take these for granted. Not just organizations have cultural blind spots; groups and individuals also have cultural blind spots. Each cultural blind spot is different depending on the direction of attention; thus, cultural blind spots across groups will to some extent be incongruent. Different directions of attention may therefore provide the opportunity for other group members to compensate in the same way that the other eye also compensates for the blind spot in the first eye. However, compensating for cultural blind spots is not automatic. Organizations need to direct attention toward such processes to be able to utilize the opportunity.

When we investigate our empirical case, we will study, in particular, the aspects of the local practices that become neglected or overlooked as cultural blind spots when attention is directed toward the global work. In this process, we will include examinations at the different locations to determine how coordination, knowledge management, and cultural practices are shaped by the global setup.

METHOD

To determine how the global setup impacts the local practices, we conducted a workplace study [20] applying ethnographical approaches [27]. Investigating GSD work practices, we took seriously the many challenges that research in distributed software development poses [26], and we conducted extensive fieldwork at the two sites involved. We studied the IT division in a Danish financial enterprise that employs more than 20,000 people in northern Europe. For confidentiality reasons, we have anonymized the company, to be referred to as *ScandiaBank*, as well as the IT division, which we here call *BankIT*. Since 2006, BankIT has been engaged in GSD by outsourcing development tasks to an IT vendor in India, *ITS* (name also anonymized). ITS is a well-reputed company working out

of Bangalore, India, with 102 years of history. It provides IT services, among other things, and ITS was originally chosen by ScandiaBank among many other IT services companies in India due to its standards in human resource management. The IT division, BankIT, comprises more than 2,000 people responsible for handling, maintaining, and developing the IT services within ScandiaBank. In India, around 750 ITS developers are associated with the BankIT, but ITS also serves other clients and customers.

The Empirical Case

The empirical case concerns the outsourcing relationship between ScandiaBank and ITS. ScandiaBank first established the outsourcing arrangement with ITS to ensure scalability of skilled labor, which was difficult to obtain onshore in Denmark in 2006. Nonetheless, a large presence in Denmark was still required to maintain business and domain-specific knowledge. However, over time the official reason for the outsourcing setup has changed to concern cost reductions. Although it is referred to as an outsourcing relationship, many of the structures resemble those of an offshore relationship where closely coupled work is required due to the distribution of knowledge and staffing across the two sites. While ScandiaBank decides which tasks and projects are dedicated to the ITS employees, and domain knowledge and business logic required for accomplishing the development work reside in Denmark, ITS is responsible for providing the needed complement of Indian employees to be associated with the bank. Because ITS possesses the required reputation in India to attract and recruit skilled labor, ITS is critical for ScandiaBank, which is not a well-known company within India and thus not prestigious enough to attract competent workers in India. Moreover, an expatriated Danish resource manager is posted in India. This individual is responsible for ScandiaBank's activities in Bangalore, together with four Danish liaison officers responsible for coordinating the different IT development projects in BankIT, sitting at the table when new employees are hired, and bridging the communication between ScandiaBank and ITS if needed.

ScandiaBank has a visual identity in terms of logo, posters, website, coffee cups, and other merchandise, all bearing the mark of a range of ScandiaBank's brand colors that are unavoidably noticeable when moving inside the walls of the BankIT building in Denmark.

In Bangalore, India, ITS is servicing different customers around the globe, one being ScandiaBank. The ITS precinct is surrounded by fences and walls demarcating ITS and the outside pulsating life of Bangalore. Clients of ITS reside in different office buildings among flourishing clean and quiet gardens. ScandiaBank has its own building where only ITS employees working for ScandiaBank are allowed access. Flags of both ITS and ScandiaBank are flying outside the entrance of the building as well as inside. The workstations are formed into cubicles and distributed into 4 large, open office environments. The walls are decorated with ScandiaBank posters in English or Danish language, and throughout the offices time zone wall clocks are hanging displaying the time in India and in Denmark. Recently the ITS employees moved into a new building, concurrent with the increase in staffing over the last seven years. The new building went through a renovation in order to meet the standards required by ScandiaBank. For example, the bank made requirements for the working environment: the colors of the office interiors needed to match the ScandiaBank brand's colors, and the ceiling height needed to be lifted.

Data Sources and Analysis

From October 2012 to March 2013, two of the authors conducted fieldwork in Denmark at a department within BankIT with a total of 80 employees and in India at their counterpart team in ITS with a total of 30 employees. More specifically, we focused on 9 BankIT employees of different occupation: system managers, IT developers, and business developers all located at one location in Denmark. From the counterpart team in India, we studied 12 ITS employees: task managers and IT developer. Furthermore, we interviewed 2 ScandiaBank expats at ITS.

Our data sources collected in Denmark and India cover 12 interviews and 80 hours of observations, as well as 12 pages of field diary, 41 pages of reflections, and 98 minutes of video-recorded accounts of our confessional statements (cf. Table 1). In total, over 300 pages of rich descriptions and interviews were created. Furthermore, internal department documents were analyzed, including outsourcing strategy documents, meeting agendas, an overview of human resources in the department, organizational diagrams, and task descriptions.

Field Site / Gathering Technique	DK	IN
Interviews (no./hours)	5/10.0	7/16.2
Observation (no./hours)	32/29.2	40/51.1
Confessional statements (no.)	8	20
Field diary entry (no.)	5	24
Reflections (no.)	60	79

Table 1. Data sources of the ethnographic fieldwork conducted in Denmark (DK) and India (IN).

In an effort to convert the data source comprising rich detailed empirical insight into concepts and premises that underlie the practices we studied, we investigated the patterns of thought and practice by combining various data-gathering techniques with a self-reflective method from action research by conducting confessional statements [4]. During the study this helped us keep track of, reflect upon, and improve the direction and inquiry of the research while being in the field. After data collection, we did several iterative write-ups, coding, and identified the critical

incidents concerning the cultural blind spots. In this work we were also guided by the theoretical framing.

RESULTS

In the following we provide background information on how the global collaboration between ITS and ScandiaBank was initiated. This includes a description of how the goals of inter-organizational collaboration were first laid out to the BankIT department. Then we present the increased extra work that emerged in the global setup. Following this, we have two main sections focusing on the local work in India and in Denmark. In each of those sections, including subsections, we present empirical data on how the local work at each site was affected by the global setup.

Initiating the Inter-Organizational Collaboration

Initially, the global collaboration was operationalized in the form of quantitative goals, which were difficult for employees to translate into concrete work arrangements and therefore failed to make the Danish employees fully committed to the collaboration with their Indian counterparts:

"When we were assigned the development organization in Bangalore there was this quantitative goal: "now everyone should have 10% of the staff in Bangalore, here you go!" [...] And it did not go very well, and we were not whole-hearted about it." (Interview, system manager, Denmark, 10/01/2012)

According to the system managers, the engagement in GSD with ITS was challenging for the people who had to collaborate on a daily basis because the IT developers themselves had to figure out suitable ways to collaborate globally without any particular guidelines from the management in BankIT.

Since 2006, though, BankIT has made efforts to provide some global collaboration guidelines; however, it is still a challenge to make the collaboration work well, and serious issues remain.

Increased Extra Work

One of the major changes when engaging in global software development work is the increased "extra work" required to handle the dependencies across time and space. For the BankIT department the increase in extra work gave rise to doubts about whether their effort in the global collaboration was a worthy investment.

Several of our informants announced that the mentality and the general assumption about the global collaboration was that it entails a lot of effort without knowing if it is actually paying off. An IT developer expressed a need for information that could help him convince people to invest time and effort in the global collaboration:

"But we have no idea what the accounts say, we have no idea whether they are paid back in two years, or 5 years. And then again take all those opinions and prejudices and skepticism that are in… amongst all, or almost all of the employees here. Concerning, well, they do not believe fully in that it can pay off." (Interview, IT developer, Denmark, 10/05/2012)

Interestingly, this *investment* is understood in two ways: 1) in terms of the extra work and time required to make it work and 2) by the business' accounts – the investment of setting up and maintaining the apparatus of outsourcing IT development to India.

Even the BankIT system manager, who is responsible for leading and motivating his staff to engage in the global collaboration, is having a hard time believing in the actual financial output of the work required:

"There's actually something that goes into production […] I do not believe that they are efficient enough yet, that we must honestly say. There is still an imbalance relative to the effort and output." (Interview, system manager, Denmark, 10/01/2012)

One reason for his misgivings is embedded in the way tasks are managed and solved in BankIT. The overhead of specifying the development tasks is, according to the system manager, a big investment, and it poses different and extra work than what they are used to in the department:

"If you pay someone to do some tasks that are incredibly well-described, i.e., down to a detail that just needs to be developed, then you have a big investment up front in describing all this. It is something we have never been very good at, and I do not think that this is the right way to do it." (Interview, IT developer, Denmark, 10/05/2012)

While documentation is a part of software development practices, the criteria for quality documentation increase in a global work setting, since remote collaborators should be able to continue work based on this documentation. The demand for quality documentation very much impacts the local IT development work of the Danish IT developers as well as the need for describing methods and processes:

"This is something I have been doing for a long time now... describing processes and so. So this has also been started up because they [the Indian IT developers] are extremely focused on following processes and templates, and doing things right. So we have to pull ourselves together and produce these things for them, right. Thing is, we know how we do things; these are simply not written down." (Interview, IT developer, Denmark, 10/05/2012)

Moving from knowing how to do things within the walls of BankIT to actually creating high-quality documentation and specifications requires different types of competences than simply expertise in programming and concomitant tacit knowledge. As such, the type of work required locally is changing and is experienced as "extra work" that emerges due to the global setup. As one developer states, the code

was not structured enough as it had been gemmating for many years:

"Just to be able to set up the apparatus to give to the Indians – it has been quite necessary. […] It has both required an effort from a business developer aspect because there must be documents, but it has required the code to be structured, and that it has not been previously. It has been, it has sprouted in ten years." (Interview, IT and business developer, Denmark, 10/02/2012)

Extra work is emerging for BankIT employees due to the global work – work that BankIT employees do not perceive as part of the work fitting their professional identity. No doubt the work is essential for global work, but there continues to be a "feeling" that this kind of work does not add directly to the production and earning of ScandiaBank and that therefore it is only a matter of time before the global setup might be closed. A system manager explains:

"We may risk that from tomorrow on someone decides that we should not use the ITS anymore, and then we cannot afford that they are the ones sitting with the knowledge, and it is obviously another parameter relative to the cooperation we can have and how much responsibility we can lay down with them when we have the little protective attitude toward our source code" (Interview, System manager, Denmark, 10/02/2012)

The changed requirements for the work by BankIT employees and the persistent doubt in the global setup, despite seven years of engagement, is a complication which makes the global work difficult to accomplish with all benefits. The lack of attention toward the complexities experienced at the Danish site has been experienced as problematic. According to the liaison officer, the Danish perspective has been de-prioritized in the past:

"One of the things we have not worked much with is the Danish employees throughout this sourcing setup. As it is no secret – it's come out to people that, yes, if in 5 years from now you're not adept at using sourcing, then you might not be here [in ScandiaBank]." (Interview, Liaison officer, India, 11/08/2012)

Clearly, the outsourcing strategy impacts the Danish employees. It changes their work assignments and creates new criteria for relevant competences and continued learning. The BankIT employees have to take on the responsibility of continuously improving their own market value through persistent education and qualification. According to the liaison officer, the employees in BankIT must be willing and ready to adapt in the future, and the work practiced in BankIT might also adapt accordingly:

"I thought about a role description… I call it 'Integration Manager' […] An integration manager is a person who can figure out how to code with the hands of others, send off some tasks, with everything it entails, which is not simple, send off some tasks, get them fixed, get them back, put them together so they can go into production. […] If you can do that, then you're still here [employed in BankIT]. However, if you want to sit and build your own LEGO castle with your own bricks then you're not." (Interview, liaison officer, India, 11/08/2012)

The message coming from the liaison officer in Bangalore is clear: the IT developer in BankIT must adapt to work in a global setting. Also, s/he must accept that the distribution of work will change in the future, moving from actual coding and programming to orchestrating and supporting the programming executed in a global collaborative setting. The global setup thus shapes the work in BankIT.

Local Work in India

In the next section we bring forward two examples: one concerning how the accomplishment of work at the Indian site is challenged by the global setup, and a second exemplifying how new management styles are introduced – shaping the local work.

Blocking or Prolonging the Work

A vital part of the work executed at the Indian site entails understanding the contextual information about the actual task that the IT developers are to solve. As we have earlier explained, the domain knowledge and the business logic reside mainly in Denmark. This arrangement means ITS employees are often waiting for BankIT employees to get back to them concerning documentation or answers to questions. In a task clarification meeting we observed from the Danish site, a Danish business developer (James) attempted to provide additional necessary information on a task to an Indian IT developer (Nikhil). Unfortunately, they both find the decisions in the task description incomprehensible:

James says he will then ask his system manager who initially made the design decision […] "I understand, there is no reason for us to go through it!" James says. It seems like he is explaining to Nikhil that he fully understands the issue; however, he is not able to give more relevant information before he has discussed this matter with the system manager. (Observation notes, Denmark, 10/04/2012)

In the above observation, the Indian IT developers are highly dependent on receiving the correct information to accomplish the tasks. Unfortunately, the originator of the proposed task and its description is the system manager, who rarely, if ever, attends these kinds of meetings due to his many other obligations as a system manager and a leader with daily HR responsibility. Thus, when sufficient information is not obtainable, the goal of clarifying a particular task is not met, and unwanted breaks then emerge. As an attempt to overcome similar situations, the Indian developers must then act proactively by continuously asking for the information whenever they are in contact with the Danish employees. At another audio meeting between a Danish business developer and two Indian IT

developers, we observed how one of the Indian IT developers seized the chance to procure some extra information about another task:

As Shashi gets to speak, it's really mostly about some other tasks than what the meeting would be about, and it is welcomed by James [...] Shashi sees a chance just to get some updates on some other issues, a kind of briefing, as well as to get some advice from James. (Observation notes, India, 10/03/2012)

The Indian IT developers are facing a lot of waiting for the Danish team to get back to them, and typically a few key actors become the bottlenecks blocking the provision of needed information. This makes it difficult to continue work, and in some situations the Indian IT developers might be lacking work, which was expressed explicitly at an internal team meeting in ITS:

Muthu says that he does not have much to do and he informs [the ITS task manager and team] that he has started looking at a new task. To that, the task manager says that he should not spend time on that [particular task] as it has not been given the go-ahead by James yet, but they are still waiting to hear what is going to happen with this task. […] Muthu, Shan, and Nikhil would soon have nothing to do. (Observation notes, India, 11/20/2012)

Evidently, what is blocking or prolonging the work among Indian IT developers is mostly related to the work of the Danish team, for example, having the Danes spending time on preparing new tasks for the Indians to solve. Here, the "extra work" required to make the global work function is missing due to key actors like "James," who was mentioned several times at the meeting and who appears to have quite a lot of work in his hands, or at least it is not a priority that several ITS employees cannot move forward because they are awaiting information.

New Management Styles

To create an environment at ITS where people in general may be more inclined to challenge leaders as well as colleagues, a mailbox has been set up for the ITS employees to be able to anonymously submit feedback or ideas. The resource manager then reads the submissions, and according to him, it is one step toward adapting an Indian top-down management hierarchy to *"our [Danish] form of management."* Additionally, an increase in employee involvement has been enacted, and the ITS employees are now involved in new initiatives such as the introduction of ScandiaBank's new strategy in Denmark. This initiative was announced through a two-hour event with stage performances by ITS employees, a red carpet, and popcorn machines, to which the resource manager commented:

"Yesterday's strategy event was an approach that hopefully will help building a sense of ownership here. In the past, this kind of information would not reach all the way down to India." (Interview, resource manager, India, 10/31/2012)

At the event, we also witnessed the resource manager trying to act more informal by wearing a tuxedo and 3D glasses, to which he told us he wanted to show that he was not so different from them. Likewise, he prefers to pick up his own coffee instead of having it served, and he makes an effort to greet people in the morning when he comes into the office.

While most changes since 2006 have concerned the Indian site, one major change influencing the Danish employees is a recent re-definition of the travel policy. Historically, only directors travelled to ITS. However, recently IT developers from BankIT started visiting Bangalore for education or/and for building collaborative relationships with their ITS colleagues. This change was also initiated to overcome some of the major resistance from the BankIT employees, as the employees visiting ITS will act as ambassadors when they return home. One of the arguments for this approach is to address the problem of making the Danish employees work globally. A liaison officer agrees that by visiting ITS in Bangalore, ambassadors for pro-global collaboration are formed:

"But it is evident that when you come down here you get something completely different, a completely different impression of what is going on. […] and there will also be some of those skeptics, as there have been many down here, and they see what the conditions are down here and how and so. And when they come home, they actually act as ambassadors to the rest of the business at home and say, 'That global collaboration could damn well make sense.'" (Interview, liaison officer, India, 11/08/2012)

Not only does this liaison officer advocate for sending employees to Bangalore to have ambassadors returning back home to Denmark, he advocates for sending the skeptics of the global collaboration. A ScandiaBank IT developer who recently visited ITS in Bangalore for 14 days also shared the understanding of what impact an onsite visit may have:

"I think it was really good, professionally speaking, to be over there, and when you come back, you can say that the cooperation continues also when you return home […] When I got home, I did not give it a thought when four Indians were writing me in one morning. Before I was like this: "Oh.. is it them again!" [laughs]" (Interview, IT developer, Denmark, 10/03/2012)

Getting to know one's global collaborators, in this case, was shown to impact the work afterward. However, what made this trip favorable was the changed perspective on the global collaboration for the Danish IT developer, namely the renewed focus of work, changing from thinking of instant messaging chat inquiries as interrupting work to becoming part of work.

Over the last seven years the collaboration between ITS and BankIT has developed and improved. However, most changes have been directed at ITS, while BankIT employees have been less of a focus. We found that the solo focus on the new remote site created problems because it neglected the important changes that also occur at the local site when a company engages in global work. This neglect, we argue, complicates the global work, and the possible benefits of global collaboration are less than what could be because successful collaboration requires engagement and participation from all partners. We will provide examples.

Local Work in Denmark

Interestingly, the local work in Denmark received much less attention than the local work in India after the global collaboration was implemented. This meant that formally local work practices in Denmark continued unchanged. However, in practice the local work in Denmark was indeed affected by the global setup. In this section we will provide empirical data illustrating this point.

Staff in the BankIT department are used to working really closely with their co-located colleagues, discussing news, urgent deadlines, or roadblocks in progressing a task by having a quick chat at the coffee machine, or by stepping into each other's offices or discussing things over lunch. Here we bring forward two examples of how work practices in the local work are challenged by the global work.

Coordinating Work with Kanban Meetings

This particular department in BankIT does not use agile development methods such as Scrum; however, they use a certain Kanban meeting mechanism as a strategy to improve translucence of the progress of tasks that people are working on in the department, as the system manager described:

"We meet every morning, just to talk about who's up to what, like, 'what kind of challenges am I facing, and is there someone who can help me?' And it has definitely given us something ... it has at least given us some visibility on what each other is doing." (Interview, System manager, Denmark, 10/01/2012)

In the field we observed a couple of these Kanban meetings executed within a local team in the BankIT department consisting of 16 employees, a combination of business developers and IT developers. At the meeting, the IT developers gathered around a large whiteboard they call a Kanban board, which shows a big chart with all their names written downward in rows and the different states of the development process presented in columns. The tasks are written down and represented by post-it notes sticking to the board. Typically, a system manager or the like would then function as moderator and each employee goes through the status of their tasks and updates the board by moving the post-its with tasks around, for example, moving a task from the state of analysis to construction.

People take turns going to the Kanban board and informing the team about the status of their tasks […] The system manager is uttering small sounds of consent, expressing that he is in on what they are talking about: "well," "yes," "yeah," "yeah well," "yes, it just sounds so," "yes, definitely," and when he is not completely following, he asks kindly, "what is it that you stand with there?" (Observation note, Denmark, 10/03/2012)

This mechanism of interaction functions as a way of communicating, coordinating, and creating translucence around what is going on. However, this mechanism does not include the global collaboration and their colleagues in ITS to the same extent as the local BankIT employees. While observing one of the Kanban meetings we noticed the limited presence of ITS, and we were told by the system manager that ITS was only represented by a single entity/row on the boards.

The system manager asked Jim about ITS because it is Jim who is the main contact […] the information from Jim was only directed to Andy [the system manager] (Observation note, Denmark, 10/03/2012)

The work executed by the ITS team and the work executed in collaboration with ITS is not visible to the same extent as the work that is performed by the employees situated in BankIT. The information from India is, in this case, mostly mediated through a single IT developer, and therefore the work of and with the Indian counterpart is not perceptible at the meetings. Acknowledging the need to create translucence around the progress of tasks being solved in India, a Danish system manager explains the rationale for now introducing a virtual Kanban board and meetings to their Indian colleagues:

"But there has just been very little managing of the tasks. And of course it is quite unfortunate because it sends out … it may send some signals to both the business but also to other developers that they are not producing anything when they've said something will be done tomorrow, and tomorrow it is still not done... and these are some of the things we have to work with; it simply needs to be more visible, possibly through Kanban […] And it is both for the business but also internally, among other developers, so they can see that there actually is progress. Something is happening." (Interview, system manager, Denmark, 10/02/2012)

By creating translucence around task progression and deliverables, the increased visibility of work of the global collaboration may lead to a better understanding of what is going on and what kind of support the global collaborative work requires. However, participation in the newly established Kanban meetings appeared to include only employees already engaged in the global collaborative work, and the board was not being merged with the Kanban board already used by the Danish team. Unfortunately, the argument of enhancing the visibility of the otherwise

imperceptible work for the customer and the IT developers not directly involved in the global collaboration is not followed up on.

Inquiry or Interruption

While in the field at the Danish site, a frustrated IT developer came to see us after we had observed an audio conference meeting with IT developers from both sites. The language barrier, he explains, is an obstacle in the collaboration, and he finds it to be easier to manage for the people who have been working with the Indian colleagues much longer than he has.

Martin says several times that it is "difficult" and that he normally tends to evade these kinds of meetings. "It is different for Allan [the Danish colleague also present at the meeting]; he talks to them every day and has done so for a year or more." (Informal talk, Denmark, 10/04/2012)

When discussing alternative communication tools he tells us that he prefers to use e-mail, whereas the Indian IT developers "are crazy about" using the internal instant messaging (IM) system provided by ScandiaBank. This obviously disparate view on the use of communication tools in the collaboration triggers a certain behavior when the Indian colleagues contact him using IM:

He explains that they do sometimes write "Hi Martin," and then they will wait for his response on the chat. The Danish IT developer finds it silly, and he would prefer if they first wrote what it was about so he did not have to respond with only a "Hi." The Danish IT developer says that he actually started to ignore these requests if they only contact him by writing him a "Hi" on the chat and without anything concrete in the message. (Informal talk, Denmark, 10/04/2012)

Interestingly, we found that chatting through IM in ITS was highly preferred, as it conforms to the physical environment of the open office cubicles by being nearly noiseless and thus less disturbing to colleagues nearby. A common work practice of using IM seems to have been adopted inside the cubicles of ITS both when communicating with co-located colleagues and when inquiring with BankIT colleagues. Unfortunately, the use of IM in BankIT is less preferred as a practice of work. Instead, we were told and observed how the employees in BankIT exchange information informally at lunch, by knocking on a co-worker's door, or by having a quick discussion in the corridor on the way to the coffee machine. Whereas IM inquiries may be perceived as interruptions, knocks on the door are regarded as part of the work and are beneficial for co-located colleagues, though disadvantageous for remote colleagues.

Summarizing our empirical results, we have pointed to how the global setup provided extra work effort in making the collaboration function well. Also, we have pointed to how the local work (both formal an informal) in India was very much shaped and re-organized based on the global setup. On the contrary, the formal work practices in Denmark were kept without aligning to the global practices. Still, we found examples where the local work in Denmark was clearly affected by the global setup, despite this being neglected by the BankIT management in their formal organization of the work.

DISCUSSION

How is the local work shaped by the global collaborative setup? First, it becomes evident that local work practices continuously transform over time as a result of the global setup. For example, the Indian local work situation is being directed toward the Danish company, both in terms of the physical surroundings (as in the logo and branding), but also in terms of mindset. An example of how the mindset of the Indian site is being directed toward the Danish company is that the Danish resource manager in India insists on picking up his own coffee or playing along during social events, which demonstrate the Danish management style. The attention toward BankIT at the Indian location was very evident by the clocks on the walls and the Indian employees' attitudes and engagement toward the Danish remote colleagues. But the changes from the global work setup were not only at the Indian location. Working in a global setup shapes all geographical sites, which includes the Danish location. The changes at the Danish site were, for example, in terms of renewed focus on methods and processes, and even more pertinent, the changing nature of what counted as "work" and "articulation work" for the IT developers in Denmark. Prior research has pointed to the malleable structures of inter-organizational and remotely distributed teams [21]. Here the argument is that over time, guided by disrupted events, reflections on the work help support the creation of new collaborative structures. We also found that working remotely created some new opportunities for reconfiguring the local practices, and as such the collaborative structures could be said to be more malleable. However, we also identified uneven attention from BankIT toward these changes. While much attention was given to the Indian location and their changes, little to no attention was given to the transformation of the Danish location. Only a few prominent changes have been made at the Danish site, for example, the travel policy that enables the Danish employees to go to India and establish a broader familiarity with their Indian colleagues. Neglecting to notice and acknowledge the changes at the Danish site related to the increase in disruptions, for example, could be interpreted either as a strategic choice by the Danish management to ignore local changes to avoid resistance to the outsourcing model, or simply lack of reflection on the matter. Studying the data in our case, we found that neglecting the changes at the Danish site was *not* a result of bad intention or a planned management strategy to avoid resistance. Instead, we will argue that the Danish location fell in the cultural blind spot [9] of BankIT. The local Danish site did not feel prepared for global collaboration and felt they were standing alone in figuring out how to

collaborate with remote colleagues. The management in BankIT had its attention on making the Indian location directed toward the business and organizational culture of the ScandiaBank [18]. This included changing the physical surroundings and management style at the Indian site. However, in this process, the local Danish site ended up in the cultural blind spot of the organization.

The global collaborative setup shapes the local work at all involved locations, both in India and in Denmark. What is interesting is that despite the many years of engagement and the continued expansion of the collaboration, the local Danish developers still articulate and experience many problems and challenges. We propose that one of the major reasons for this diverse perspective on the collaboration concerns the asymmetry in attention from BankIT toward the collaborative setup, where the Danish site is in the cultural blind spot. However, important changes occurred at the Danish site as a result of the global setup.

One of the major changes concerned the nature of the work for the IT developers at the Danish site. Collaboration consists of work and articulation work, which are seamlessly integrated in practice [28]. Still, the distinction between what counts as work and what counts as articulation work is important for the professional identity of different organizational members. Prior to the global setup, the Danish IT developers had their professional identity as technical experts focusing on system architecture, programming, and highly sophisticated technical knowledge; however, the global setup changed this, and they started to do all the "extra work." The IT developers in our empirical case clearly referred to a distinction between their work and the extra work required to work in a global setup. This "extra work" was the articulation work for global collaboration. Danish IT developers found themselves spending much time on tasks related to methods and formal processes, which to some extent were perceived as an external demand that constrained the actual work. We are aware that methods and formal processes for some might be viewed as "the work" of IT developers [16]. However, in our empirical case, the IT developers in Denmark did not. Prior to the global setup the Danish IT developers perceived the close, friendly relationship they had with local colleagues and which was accomplished through the daily face-to-face encounters to be directly beneficial for supporting collaboration, and all additional time spent on methods and processes was a waste of time. In the global setup, this balance changed and the Danish IT developers now had to submit to methods and formal processes due to the global engagement, where their strategy previously had been to ignore such external requirements. In addition, it was clear that all business as well as system integration knowledge was located at the Danish site, which meant that extra time was required by the Danish IT developers to communicate frequently with their remote counterparts, reducing the time to be spent on programming. The global setup thus impacted the local work for the Danish IT developers by increasing their efforts invested in articulation work while reducing the time for actually developing code. You might argue that the extra work required for communication and coordination simply became part of the Danish IT developers' work – their formal job description; however, as long as their professional identity perceived communication, processes, and methods as outside their core work, this change created tensions. What counts as work or articulation work, and for whom, is dependent upon the professional role of the collaborators. In a group of globally distributed IT developers, who share a professional context, what counts as the *real* work is often related to the activities of writing code, working with business and system requirements, conducting tests, etc. In these cases, articulation work concerns, for example, the coordination of activities and dependencies. When working globally, the increased reach of the activity [12] requires extra effort in handling articulation work. Therefore, the effort involved in handling articulation work is more time and resource demanding on the participants. This increased time spent on communication in global software development has also been reported elsewhere [6, 17]. However, our argument here is that this shift in the balance between work and articulation work is a more fundamental debate about the kind of competences and qualifications that make up the job descriptions for IT developers working in global settings. It is no longer enough to have high technical qualifications; instead, qualifications now include global work competences such as communication and coordination, and this is part of the work required to succeed in collaborating globally. The boundaries between what counts as work and what counts as articulation work are re-defined for the IT developer. For the IT developer, articulation work now becomes part of the formal work required when participating in global work; it is no longer *extra* work.

Coordination is a strategy to handle the extra effort of articulation work [12]. Coordination has typically been organized through standards, protocols, and coordination mechanisms [29], or in terms of timely coordination [23, 24]. The increased effort spent on articulation work also requires increased focus on coordination practice. Part of the extra effort concerns applying formal processes and methods to create standards for work across sites. However, we also found that a critical change of the global setup concerned the prioritizing of tasks. Prior to the global setup, the IT developers did the prioritizing of tasks and activities autonomously through face-to-face discussions or simply by making decisions and conducting the task. If colleagues experienced issues or problems and need help, they would simply go to other colleagues who then would prioritize to spend time supporting each other. Interruptions were not seen as disruptive but as a part of the normal, natural trouble [27]. In the global setup, the number of colleagues increases, as does the number of colleagues who need to interrupt to solve issues or questions. In our case, this

situation was further increased by business knowledge only being present at the local Danish site. People do not physically knock on the door in a global setting; moreover, awareness of whether remote colleagues are working toward a tight deadline or are busy is also not automatically available in the global setting [14]. Instead, "knocking on the door" will manifest as digital interruptions through instant messaging or similar tools. However, whether interruptions are perceived as disruptions or normal, natural trouble is very dependent upon the relationships between colleagues, as well as the type of inquiry. If inquiries are perceived as questions that a colleague could get answered by other means (e.g., reading documentation) and are thus not important enough to warrant interruption, then the interaction will be problematic. In our case, it was evident that employees at the Danish site experienced increasing interruptions from their remote colleagues. However, there were clear differences in how these interruptions were perceived. When the Danish IT developers knew and had a relationship with the remote colleagues, they would engage in knowledge work [2], seeing such interruptions as though local colleagues were knocking on their door, and would provide the answer in a timely manner, also taking into account the time zones. In addition, they knew more about each other's work [25] and would be aware of situations where remote colleagues could not move forward before getting an answer, thus prioritizing such inquiries. In contrast, Danish IT developers without close relationships with the remote colleagues would view each interruption as a disruption that required an instant response and, as a strategy, choose to ignore remote colleagues "knocking on the door." In other words, the asynchronous use of instant messaging [7] was not considered as an option for some developers. Further complicating the matter in our case was that the Danish system manager took great interest in technical details of the solutions yet was very busy and difficult to get a hold of, which in several situations disrupted the project with unwanted breaks. Moreover, these unwanted breaks were also a product of the challenges of equal motivation for knowledge exchange across sites [13] and the dynamics of sub-groups [10]. All domain experts were at the same location, and the IT developers solving the actual programming tasks were at another. This division of expertise in concurrence with the geographical location increased the us/them dichotomy, forming apparent sub-groups between domain experts at one location and IT developers who were solving the actual programming tasks at another location. The competences of coding "with the hands of others" includes timely coordination and frequent interaction based on relationships among remote colleagues, taking into account time zones and ensuring that unwanted breaks do not occur. While previous research has pointed to the importance of frequent and timely interaction [e.g. 23, 24], our argument here is that the global setup impacts the local setting by increasing the number of colleagues who need to interrupt the work of others in order to do their own work, and as such the increase in interruptions and acting upon these must be part of the employees' work and professional identity. We found that this change due to the global setup impacted the work at the local Danish site but was outside the attention of BankIT.

Translucence in communication structures is critical for global work [5]. Remote colleagues need to make their work visibly available for others to monitor to create accountability. When the work becomes increasingly dispersed between participants as well as across geography and time, bringing translucence into the work practices becomes essential. In our case, it was very evident that the remote work of the Indian colleagues was absent in the everyday work of most of the Danish IT developers. Only the few Danish employees who had daily contact with their remote colleagues knew and were aware of what was going on remotely; however, the majority of the Danish IT developers had no relationship with their remote colleagues. This was evident, for example, in the Kanban board, where the work of the entire Indian counterpart team was reported briefly by one of the Danish IT developers. The remote colleagues were visualized in a single row on the board, whereas each Danish colleague had his or her own row. The invisibility of the remote colleagues created extra work for the few who worked closely with the developers in India because all interactions became mediated by them and translated to others; however, all the work required to continuously make visible in the Danish setting the huge amount of work that was conducted in a global setup was an add-on work task. The invisibility also made the Danish employees (not working closely with their counterparts in India) even more alienated from the global setup. The lack of translucence of the global work among the larger group of IT developers in Denmark had further impact by reducing chances for them to create relationships with their remote colleagues, which were essential for perceiving the necessary interruptions as part of the everyday work rather than as disruptions.

By directing all attention toward the Indian local site, the management of BankIT neglected to notice the changes in the local work at the Danish site. The Danish location fell into the cultural blind spot of management. By neglecting attention toward the changes at the Danish site, it created obstructions for the global work in general, particularly related to the Danish IT developers' perception of interruptions, as well as to the nature of work. The cultural blind spot of the Danish organization thus affected the ability to take a critical stance toward its own practices, and thus hindered the development into a global organization.

CONCLUSION

In this paper, we presented empirical observations from our ethnographic study of global software development between Denmark and India, demonstrating how a global setup impacts the local work practices of both sites. We found that the Danish outsourcing company in our case focused only on the Indian site of the collaboration, while

lacking attention toward the changes in their own organization and the local work conducted by Danish IT developers. We refer to this lack of attention as a cultural blind spot. Due to the cultural blind spot, tensions were created in the global setup, influencing the collaborative work practices between remote colleagues.

Cultural blind spot serves as a helpful concept when seeking to open up critical issues of global cooperative work in CSCW research. The cultural blind spot is the information and practices embedded within our own cultural behavior and practices that we take for granted and thus risk neglecting. By introducing the notion of cultural blind spots to CSCW, we argue that there is a need for organizations to compensate for their blind spots by making visible the critical issues in the local work that otherwise go unnoticed. In our empirical case, we identified three local issues that fell into the cultural blind spot of the Danish organization: 1) the increased number of interruptions, 2) the lack of translucence of remote colleagues' work, and 3) the re-definition of the boundaries between what counts as professional work and what counts as articulation work for the Danish IT developers.

The increased interruptions that the IT developers experienced based on the increased number of colleagues to collaborate with (locally and remotely) were overlooked and thus disregarded as an important issue to engage with by management because it fell into the cultural blind spot of the Danish company. Still, the increasing interruptions were an enormous change that the outsourcing relationship brought to the home ground of the Danish company. The impact of transitioning from working with software development locally to globally showed when the Danish company failed to notice the extra work required to integrate the coordination of remote work into the local coordinative artifacts like the Kanban board at the Danish site. Here it was clear that the global setup was lacking translucence because of the Danish participants' cultural blind spot that caused them to overlook the necessity of representing the day-to-day work and coordination with their remote colleagues on the board. This oversight had a significant negative impact on the global collaboration because the articulation work required to handle the global collaboration became opaque and the people handling it were not given credit for their efforts.

It is evident from our study, also supported by previous studies [e.g. 12], that the effort required to handle articulation work increases in the global setting. As it is, the participants involved in GSD do articulation work to make the work work. However, in this paper we showed how the increase in effort involved in handling articulation work required to make GSD function better tends to be neglected. We argue that the increase in effort required to handle articulation work for global collaborative work is inevitable. It also changes the balance between the efforts and resources that an IT developer involved in software development spends on programming and testing, for example, and the efforts that same individual spends on articulation work to make the collaboration function well. In this way, our study suggests it is important to re-define what counts as work and what counts as articulation work in the formal descriptions of IT developers' work when they become involved in global software development. Thus, we argue that global software development practices change the qualifications and competences involved in being an expert IT developer in global collaborative work. Working in a global setting changes the conditions for IT development, and we need to figure out how we can support the expert IT developers, who have the critical business knowledge to better work in global setups, to *"figure out how to code with the hands of others"* while developing their own skills and continuously improving their own professional profile. Actions must be taken to address organizations' cultural blind spots, to solve the inability to take a critical stance toward one's own practices, and to continuously make changes that go along with becoming a global organization.

ACKNOWLEDGEMENT

This study is conducted within the context of the interdisciplinary and inter-organizational research project "Next Generation Technologies and Processes for Global Software Development," #10-092313 a research project that is financially supported by the National Council for Strategic Research, Ministry of Science, Innovation and Higher Education in Denmark. For further information, see www.nexgsd.org. We would like to thank all the anonymous volunteers from the global team in ScandiaBank and ITS who generously offered their time and energy for this research. Finally, we would like to thank Morten Esbensen, Steven Jeuris, Anne-Marie Søderberg, and Rasmus Eskild Jensen for valuable discussions about the study, and the anonymous reviewers for insightful comments and suggestions.

REFERENCES

1. Al-Ani, B., et al. (2013). Globally distributed system developers: their trust expectations and processes. Proceedings of the 2013 conference on Computer supported cooperative work. ACM**:** 563-574.
2. Avram, G. (2007). Knowledge Work Practices in Global Software Development. *The Electronic Journal of Knowledge Management*. 5(4): 347-356.
3. Avram, G., et al. (2009). Bridging, Patching and Keeping the Work Flowing: Defect Resolution in Distributed Software Development. *Computer Supported Cooperative Work (CSCW)*. 18(5-6): 477-507.
4. Bjørn, P. and N. Boulus (2011). Dissenting in reflective conversations: Critical components of doing action research. *Action Research*. 9(3): 282-302.

5. Bjørn, P. and O. Ngwenyama (2009). Virtual team collaboration: building shared meaning, resolving breakdowns and creating translucence. *Information Systems Journal*. 19(3): 227-253.
6. Boden, A., et al. (2009). Knowledge Management in Distributed Software Development Teams - Does Culture Matter? In *Proc. Conference*, IEEE Computer Society: 18-27.
7. Boden, A., et al. (2007). Coordination Practices in Distributed Software Development of Small Enterprises. Proceedings of the International Conference on Global Software Engineering. IEEE Computer Society**:** 235-246.
8. Boden, A., et al. (2009). Trust and Social Capital: Revisiting an Offshoring Failure Story of a Small German Software Company. ECSCW 2009. I. Wagner, H. Tellioğlu, E. Balka, C. Simone and L. Ciolfi. Published, Springer London**:** 123-142.
9. Clausen, L. and M. Zølner (2010). Cultural Awareness Processes in Corporate Knowledge Sharing: Turning "Cultural Blind Spots" into Resources. Paper presented at The Academy of Management Annual Meeting 2010, Montréal, Canada.
10. Durnell Cramton, C. and P. J. Hinds (2005). Subgroup Dynamics in Internationally Distributed Teams: Ethnocentrism or Cross-National Learning? . *Research in Organizational Behavior*. 26(0): 231- 263.
11. Faraj, S. and L. Sproull (2000). Coordinating Expertise in Software Development Teams. *Manage. Sci*. 46(12): 1554-1568.
12. Gerson, E. M. (2008). Reach, Bracket, and the Limits of Rationalized Coordination: Some Challenges for CSCW. Resources, Co-Evolution and Artifacts. Published, Springer London**:** 193-220.
13. Gumienny, R., et al. (2013). Supporting creative collaboration in globally distributed companies. In *Proc. Conference*, ACM: 995-1007.
14. Herbsleb, J. D. (2007). Global Software Engineering: The Future of Socio-technical Coordination. 2007 Future of Software Engineering. IEEE Computer Society**:** 188-198.
15. Hossain, E., et al. (2011). Scrum practices in global software development: a research framework. In *Proc. Conference*, Springer-Verlag: 88-102.
16. Iversen, J. H., et al. (2004). Managing risk in software process improvement: an action research approach. *MIS Q*. 28(3): 395-433.
17. Jensen, R. E. and P. Bjørn (2012). Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds. From Research to Practice in the Design of Cooperative Systems: Results and Open Challenges. J. Dugdale, C. Masclet, M. A. Grasso, J.-F. Boujut and P. Hassanaly. Published, Springer London**:** 123-136.
18. Krishna, S., et al. (2004). Managing cross-cultural issues in global software outsourcing. *Commun. ACM*. 47(4): 62-66.
19. Latour, B. (2005). Reassembling the Social: An Introduction to Actor-Network-Theory, OUP Oxford.
20. Luff, P., et al. (2000). Workplace Studies: Recovering Work Practice and Informing System Design, Cambridge University Press.
21. Majchrzak, A., et al. (2000). Technology adaption: the case of a computer-supported inter-organizational virtual team. *MIS Q*. 24(4): 569-600.
22. Marlow, J., et al. (2013). Impression formation in online peer production: activity traces and personal profiles in github. Proceedings of the 2013 conference on Computer supported cooperative work. ACM**:** 117-128.
23. Massey, A. P., et al. (2003). Because Time Matters: Temporal Coordination in Global Virtual Project Teams *Journal of Management Information Systems*. 19(4): 129-155.
24. Maznevski, M. L. and K. M. Chudoba (2000). Bridging Space Over Time: Global Virtual Team Dynamics and Effectiveness *Organization Science*. 11(5): 473-492.
25. O'Neill, J., et al. (2011). When a little knowledge isn't a dangerous thing. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM**:** 1667-1676.
26. Prikladnicki, R., et al. (2013). Data collection in global software engineering research: learning from past experience. *Empirical Software Engineering*: 1-35.
27. Randall, D., et al. (2010). Fieldwork for Design: Theory and Practice, Springer.
28. Schmidt, K. and L. Bannon (1992). Taking CSCW seriously. *Computer Supported Cooperative Work (CSCW)*. 1(1-2): 7-40.
29. Schmidt, K. and C. Simone (1996). Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *Computer Supported Cooperative Work (CSCW)*. 5(2-3): 155-200.
30. Søderberg, A.-M., et al. (2013). Global Software Development: Commitment, Trust and Cultural Sensitivity in Strategic Partnerships. *Journal of International Management*. (in press).
31. Strauss, A. (1988). The articulation of project work: An organizational process. *The Sociological Quarterly*. 29(2): 163-178.
32. Walsham, G. (2002). Cross-cultural software production and use: a structurational analysis. *MIS Q*. 26(4): 359-380.

Paper 2

Matthiesen S, Bjørn P. Why replacing legacy systems is so hard in global software development: An information infrastructure perspective. New York, New York, USA: In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15). ACM, New York, NY, USA, 876-890.

Why Replacing Legacy Systems is So Hard in Global Software Development: An Information Infrastructure Perspective

Stina Matthiesen
IT University of Copenhagen
Rued Langgaardsvej 7,
2300 Copenhagen,
Denmark
matthiesen@itu.dk

Pernille Bjørn
IT University of Copenhagen,
Rued Langgaardsvej 7, 2300 Copenhagen, Denmark
UCIrvine & Intel Center for Social Computing (ISTC)
Bren Hall, 6th Floor, Irvine, California, USA
pbra@itu.dk

ABSTRACT
We report on an ethnographic study of an outsourcing global software development (GSD) setup between a Danish IT company and an Indian IT vendor developing a system to replace a legacy system for social services administration in Denmark. Physical distance and GSD collaboration issues tend to be obvious explanations for why GSD tasks fail to reach completion; however, we account for the difficulties within the technical nature of the software system task. We use the framework of information infrastructure to show how replacing a legacy system in governmental information infrastructures includes the work of tracing back to knowledge concerning law, technical specifications, as well as how information infrastructures have dynamically evolved over time. Not easily carried out in a GSD setup is the work around technical tasks that requires careful examination of mundane technical aspects, standards, and bureaucratic forms, as well as the excavation work that keeps the information infrastructure afloat.

Author Keywords
Global software development (GSD); outsourcing; ethnographic study; information infrastructure; legacy systems; interface integration; system interfaces; excavation work

ACM Classification Keywords
K.4.3 [Organizational Impacts]: Computer-supported collaborative work; H.5.3 [Group and Organization Interfaces]: Computer-supported cooperative work

INTRODUCTION
Today, working remotely is the norm for how software development is accomplished [25]. In this paper, we study a particular type of software development, namely global software development (GSD), which refers to the development of large-scale software systems organized in globally distributed organizations with IT developers located across different time zones and geographical locations and using IT-enabled collaboration tools [27]. Expanding IT development activities to remote collaborators unquestionably poses additional challenges to collaboration – challenges that are also directly relevant to computer-supported collaborative work (CSCW) research [5]. Previous research has therefore investigated articulation work [9, 35], coordination [8, 13], routine [21], delay [26], knowledge [7], and awareness [16]. Additionally, cross-cultural issues [28, 32], vendor expertise and location [1], and the nature of the collaborative partnership that needs to be established [43] are all considerations to be discussed. Thus, we see research communities and conferences creating a growing a body of literature on GSD [2].

Despite the increasing interest in and various studies of GSD in CSCW, there seems to be a tendency to predominantly report on difficulties related to collaborative work, *without* considering the nature and characteristics of software systems involved in the work task. This paper seeks to unfold the nature and characteristics of the software task, determining how it provides certain conditions for the collaborative work in GSD setups.

The software task we investigated is a specific type of task: the replacement of a legacy system. The development of software systems does not emerge from scratch; rather, new systems must be built to replace older systems containing obsolete technologies, also referred to as legacy systems. Legacy systems are systems performing critical business functions essential or useful to an organization. Legacy systems tend to have long lifespans, as they often are too expensive or critical to replace [44]. In this paper, our interest is to focus on the actual work task of replacing a legacy system in order to understand its nature and complexity. We aim to determine why such tasks place constraining and complex challenges on GSD work practices, increasing the risk of failure in collaboration. The work task in GSD is a collaborative task by definition [42] since multiple individuals are required to participate and are thus dependent on each other. Each individual task changes

CSCW '15, March 14 - 18 2015, Vancouver, BC, Canada.

ACM 978-1-4503-2922-4/15/03…$15.00
http://dx.doi.org/10.1145/2675133.2675232

the state of the common field of work, meaning collaborators must act and react accordingly [42].

We report on findings from an ethnographic study on GSD practices conducted between June 2013 and December 2013. The study involves a Danish IT company (MData) engaged in collaboration with an Indian IT vendor (ITech) that won a public tender competition from the Danish government to develop a new system to replace a legacy system that was a core component of the Danish governmental IT infrastructure. In the case of MData and ITech, MData primarily chose to outsource the development of the technical interfaces where no direct users were involved, and it was therefore assumed to be a straightforward task. However, as it turned out, the task was anything but straightforward, and they failed to have it outsourced. In the end, MData had to leave the interface tasks to the Danish developers, who then redid the system development completely. This collaboration failure had less to do with communication, coordination, or cultural complexities – rather, it had to do with the complexity of the task. What made it so difficult to develop the technical interface in the GSD setup? What were the characteristics of the software task replacing technical legacy systems that increased the complexity in solving the task in an outsourcing collaborative setup?

This paper has a three-fold contribution. First, we offer a detailed empirical account of the characteristics of the work task involved in replacing legacy systems that are part of the core IT governmental infrastructure. Second, we point to how replacing legacy systems within governmental setups is *not simply* a technical and pre-defined task, but instead requires a significant amount of work in tracing to the source of information and knowledge concerning both legal and technical specifications, as well as in understanding how the information infrastructure has dynamically evolved over time. Finally, we demonstrate how the framework of information infrastructure can help illuminate the intertwined and complex work involved in replacing legacy systems within governmental IT systems supporting interconnected social services within modern digital societies.

The rest of the paper is outlined as follows: First, we present related work, followed by a description of the research method in presenting the empirical case that accounts for our research approach and data analysis method. Second, as we approach an answer to our research question, we then present the empirical results, highlighting the complex nature of the particular development work we are examining. Third, we zoom out from our findings and discuss how the empirical examples together depict the complexities of GSD when developing an IT system to replace a legacy system, specifically within the governmental IT systems of Denmark. Finally, the paper concludes.

RELATED WORK: LEGACY SYSTEMS & INFORMATION INFRASTRUCTURES

Legacy Systems

Legacy systems are informally known as "large software systems that we don't know how to cope with but that are vital to our organization" [3, p.19]. Legacy systems are socio-technical systems situated within an organization that include people, software, and hardware. These systems have often been around for a long time and are built on now obsolete forms of technology, thus requiring constant maintenance. Legacy systems perform critical business functions that are essential or useful to an organization, which means they often have long lifespans as they are too expensive or critical to replace [44]. Software evolves over time, and in the majority of legacy systems the original structure has disappeared and up-to-date documentation is lacking or non-existent; thus, the most reliable information about the system is the actual source code. In these instances, legacy systems represent years of accumulated experience and knowledge about the organization, and may be the only places were business rules exist [3]. Research on legacy systems is mostly technical in nature, proposing different strategies for how to handle legacy systems in terms of reverse engineering or refactoring [33]. However, the most recent research shows that the challenges of legacy systems are not only technical, but also business-related and organizational [30].

When developing large software systems, there are great challenges in handling and managing dependencies among systems/modules and the developers' work. One of the most influential principles in software engineering is Parnas' idea of modular decomposition and *information hiding* [38]. This involves breaking problems into smaller pieces, allowing people to work on a software module in a more isolated fashion with decreased need for communication and coordination. As an example, this is instatiated through interfaces, which are means of organizing a program as a set of modules that can communicate with one another, with clear specifications of procedures and accessible variables [14, p.179]. Modules are then able to call for and use another module's functions while the actual implementation remains hidden [22]. Interfaces are sometimes called Application Programming Interfaces (APIs), which are pre-defined source code materials with clear documentation of existing programs or sub-systems. These offer a range of services that can be accessed by calling interface procedures in specific ways [44, p.135].

The advantages of Parnas' principle are commonly agreed upon among practitioners and researchers. This principle states that, in facilitating independent IT development work, it is useful to conduct IT development in parallel to separate modules by reducing technical interdependencies among modules, thus creating loose coupling between the tasks. But this principle has been found to be extremely difficult to follow by several CSCW and software engineering scholars. For example, Cataldo and Herbsleb

stress "the importance of understanding the dynamic nature of software development" [11, p.8] and propose that the *"right"* set of technical dependencies needs to be identified to determine the relevant work dependencies and to coordinate accordingly. This proposition is based on a conceptual framework of sociotechnical congruence that examines the relationship between a software system's technical dependencies and the task structure of the development work [12]. Moreover, the assumption that interfaces or APIs reduce the need for coordination and communication among software developers has been questioned. For instance, de Souza and his colleagues noticed that working with APIs has side effects on collaborative software development among teams since labour is likewise divided [45]. Furthermore, it is argued that interfaces alone do not induce developers to have less coordination needs [15]. Instead, APIs are enacted in similar ways to contracts, drawing up the boundaries of developers' individual work. APIs also work as devices for communication in the coordination of software developers' work. Therefore, in order for APIs to benefit from their advantages, workarounds are sometimes required. Proper coordination and communication engagement is also required to ensure the alignments of the integration work among the software developers using or implementing the APIs [15].

Nevertheless, the previously mentioned implications have received far less attention than the technical advantages of Parnas' principle. In response to the issues of decomposition and division of labour, Grinter [22] points to recomposition, which refers to all the work necessary to coordinate the dependencies among components. This is the critical body of work that entails reconnecting the software component to a whole system. Through recomposition, aspects of software development work are revealed that include problems caused by waiting for external or internal vendors or development departments to deliver hardware, software, or organizational resources.

Another type of interface, called system interfaces, is a means of organizing communication across various sub-systems. It translates diverse types of inputs from the systems either into outputs that can serve as inputs in other systems or into compared data, producing new types of data for outputs. System interfaces are critical when replacing legacy systems, since new sub-systems have to fit into the existing information infrastructures through these interfaces. The classical software engineering book by Sommerville [44] clearly states that if the "new system and the existing systems must work together, the interfaces of existing systems have to be precisely specified" [44, p.135]. Moreover, it is suggested that such important specifications of system interfaces must be defined early in the process and included in the requirement documentation [44]. However, this is exactly the problem with legacy systems. No formal documentation exists and the years of sprouted source-code, fixes, and add-ons make it impossible to specify the system interfaces in advance.

We join the previously mentioned scholars in investigating the collaborative limitations provided by interfaces and APIs in software development work. But our approach differs from previous software engineering research because we move further into the nature of the concrete development task. We seek to explore how this task materializes within dependencies of a less technical, but more sociocultural, character. Therefore, in this paper we want to investigate the collaborative challenges and complexities involved in replacing legacy system interfaces that are part of a larger governmental information infrastructure.

Information Infrastructures

In unpacking the nature of the IT development task when replacing legacy systems, we turn to the information infrastructures literature, which originated from Star and Ruhleder [47]. Bowker and Star [10] define various characteristics of an infrastructure. For example, an infrastructure is embedded into other structures, technologies, and social arrangements; it transparently supports tasks, within a spatial or temporal scope; and it is shaped by and shapes conventions of practices. Moreover, an infrastructure is also modified by the embodiment of standardized tools and other infrastructures. At the same time, it grows from, inherits strengths and limitations, and wrestles with the inertia of the installed base from which it is built upon. An information infrastructure is large, layered, and complex. It also plays different purposes and means different things to different groups, depending on locations. Finally, it only becomes visible when it is breaking down [46]. Thus, changing infrastructures takes time, requiring negotiations and adjustments with other aspects of the involved systems. Information systems are part of what makes the information infrastructure, and are constructed on the basis of pre-existing work, systems, and practices. The information systems have dynamically evolved over time and are thus part of what shapes the information infrastructure. In this way, pre-exisiting information systems place conditions on the degree of freedom that can be exercised by bringing in new IT systems and practices [37].

In CSCW, information infrastructures literature is mostly known within the area of scientific research exploring distributed collaborative scientific research in terms of cyberinfrastructure (in the United States), e-Science (in the European Union) or eInfrastructure [see e.g. 18, 24, 41]. This type of research seeks to identify what is referred to as scientific information infrastructures, which make up "a specific class of infrastructure that brings together people, information, and technologies to support research" [4, p.245]. Research on scientific information infrastructures investigates how the development of scientific infrastructures seeks to support and benefit from the sharing

of resources, knowledge, and data in order to enable new forms of investigations and the emergence of novel science [41].

Information infrastructure research on scientific work has made important contributions to the CSCW community over the last few years [49]. Scholars have focused on various aspects of infrastructural scentific work, such as the design of an infrastructure with long-term resources and where goals of embodying reuse and stability are pursued [40], the role of embeddedness in infrastructure development [4], and the issues of scaling an infrastructure for potential new users [49], to mention a few. While our focus in this paper has little to do with leveraging large-scale distributed practice in the sciences or the challenges of how collaboration is constrained or enabled through the nature of an infrastructure, our use of the information infrastructure approach shares important characteristics. Namely, that information infrastructure is fundamentally a relational concept that only reveals itself through practice, when the ways in which the infrastructure serves certain practices while constraining others become apparent [46]. When investigating larger scale technical systems being developed, it is difficult to point exactly to what the infrastructure consists of; however, it holds that certain dimensions [10, 47] – such as embeddedness, transparency, and embodiment of standards – become visible and fixed in modular increments upon breakdown [46]. Thus, by taking an information infrastructure perspective, as has been used within the studies of scientific work, when exploring the difficulties in replacing a legacy system in the GSD setup, we explore the legacy system as the inertia of the installed base, which is part of structuring the large information infrastructure of the Danish government.

Information infrastructures have been developed as part of a theoretical lens to explore the basic nature and complexities of larger IT systems within healthcare, particularly in Scandinavian countries [6]. Based on the work of Star and others, this set of information infrastructures research seeks to understand the complex IT landscapes that make up healthcare systems in general and, more specifically, hospital systems [19, 23]. The interest is to understand how and why it is so difficult to design and implement large-scale systems supporting governmental healthcare policies that enable the exchange of information across institutions within Scandinavian countries [36]. This has led to discussions between standardization and flexibility [24]. The general lesson from this literature is that IT systems within hospitals are the collective result of patchwork integration of multiple sub-systems [20], each the result of modular incremental changes without any one in particular being "in charge" of the whole.

Interestingly, there are many similarities between governmental hospital systems and social services systems, the latter being the interest in this paper. Thus, in this paper, we apply the theoretical lens of information infrastructures as it has been used in CSCW, as well as in healthcare system research, to study the complexities in replacing legacy technical interface systems in a GSD setup.

METHOD

In order to investigate the complexities of GSD work and the nature of the software development task from a sociocultural angle, we chose to follow a GSD project that was assumed to be both technically complex and highly entangled in sociocultural dependencies. Through an intensive workplace study [34], we investigated the challenges in GSD work and work practices in a particular GSD setup between Denmark and India. Over the course of 2013, we collected ethnographic data [39] at multiple field sites.

The Empirical Case

The field sites included a Danish IT company, here called *MData*, and their global IT supplier in India, here called *ITech*. MData is a large IT and software company with more than 3,200 employees and several branches around Denmark. For more than 40 years, MData has been developing software systems that support the Danish public sector in handling welfare benefits and various administrative systems. MData develops IT solutions for the government and private markets, and one of their core competencies is SAP programming – a standard Enterprise Resource Planning (ERP) system used for managing a business.

From both the Indian and Danish sites, we followed a GSD IT project developing an illness benefit payment system to support Danish municipalities in the administration of public welfare benefits for their citizens. We identify the project as the *SickPay project* developing the *SickPay system*. This new system has to conform to and partly replace some functionality in the current legacy system, a 30-year-old system called *WBenefit*, used by the caseworkers employed in Danish municipalities to administer various citizen welfare benefits, including unemployment benefits, sickness cash payments, and maternity pay. The WBenefit legacy system interlinks with various sub-systems through a range of system interface connections in order to enable communication across the range of Danish government IT systems. The work of replacing some of the functionality from the WBenefit system to the SickPay system therefore entails the implementation of similar system interface connections. Moving forward in this paper, we use *integration* as an emic term for the work of bridging the new development with the legacy code.

MData works on several types of IT projects, including those based on government contracts offered to both the government and the private sector, and MData's own projects developed to extend, change, or maintain current IT solutions or, alternatively, to develop new off-the-shelf products. The WBenefit system is also maintained by a group of MData people – the WBenefit team consisting of

eight employees with technical, maintenance, and business-oriented job roles. The WBenefit team knows a lot within the domains of handling welfare benefits, including unemployment benefits, sickness benefits, and maternity allowances in the WBenefit system and other connected IT systems. Similar to the SickPay project, MData is also working on a *MaternityPay* project to develop a *MaternityPay* system to support allowances for parental leave. Since 2006, MData has been engaged in GSD offshore projects with the Indian IT services company ITech to offer customers a reduced time to market and ensure extra resources and growth within the SAP development area. Today, MData employs more than 200 people within five different global suppliers located in Poland and India, with ITech being the largest.

Nearly 60 employees are working on the SickPay project from different sites. This includes three solution teams at MData in Denmark and one partially dispersed team – referred to as the *offshore team* – that consists of six ITech employees in India, five MData employees in Denmark, and one ITech employee and one external application consultant located onsite at MData in Denmark. The project is an agile software development project that progresses in time-limited iterations of five weeks each**,** or sprints**,** during which an outline of development tasks is first planned and estimated, then implemented, and finally reviewed/evaluated. Each of the four solution teams has a scrum master and a tester, as well as several IT developers, application consultants, and business specialists who understand the legislation. Business analysts build bridges between the legislation and the system through their knowledge of how the solution could best be implemented within the utilized technology. In addition, the project has a steering group, an end-to-end project manager, a Quality Assurance (QA) test team, a Product Owner & Architecture (POA) team, and a shared service team, all of which are responsible for completing the trajectory between the development and testing phases. The development tasks – the Product Backlog Items (PBI) – in the project are initially assigned to the solution teams by the POA team.

The offshore team, consisting of employees from both India and Denmark, "meet" and talk at least once a day at their online daily scrum meetings (booked for 30 minutes each), using shared screens and audio conferencing systems run from the team members' own desktop computers. Across the two sites, business analysts and testers are located in Denmark, while the core team of IT developers are located in India. The collaboration and work on a task would typically flow as described in the following example. The whole offshore team is assigned a number of tasks (PBIs) at the beginning of a sprint and these are kept in a shared document on SharePoint. When assigned a PBI, the business analysts start going through the PBI to acquire knowledge and information in order to create business refinement documents that further detail the development task. These documents are to be handed over to the IT developers. When the refinement documents – which could include drawings of business processes and interface descriptions – are ready, the business analyst schedules an online audio conference meeting with the relevant IT developers and testers, and then hands over as much as s/he knows about the given PBI. The initial meetings can be quite lengthy (lasting up to several hours), depending on the complexity of the task. Following this meeting, the IT developers begin the development and the testers start writing the test cases. At the daily scrum meetings, each team member gives updates on his or her progress and status on current work, which sometimes triggers new meeting sessions to be scheduled, if a given PBI needs further clarifications. At the daily scrum meetings, the scrum master will follow up on how work is flowing and continually assign new PBIs to the team members. At the end of a sprint, work is reviewed and retrospectively PBIs that have not been finalized may be taken into account when the next sprint is planned before launch.

Data Collection and Sources

Field Site / Gathering Technique	DK	IN	Online
Interviews (no./hours)	10/8.8	7 /5	4/2.6
Observation (no./hours)	29/35.6	17/9.1	26/14.3
Field diary entry (no.)	39	9	-
Time spent in field (hours)	137	66	-

Table 1: Data sources of the ethnographic fieldwork conducted in Denmark (DK) and India (IN). Observations and interviews conducted online are also represented in the table.

The data were collected applying ethnography observations in both Denmark and India. Observations entailed shadowing the workers in their daily work and listening in on calls and online meetings, interviews, or document analyses. Various data gathering techniques were used to allow for triangulation of results, including: semi-structured interviews of both a formal and informal nature, observations of various work activities, and retrievals of documentary evidence. The interviews were conducted in face-to-face meetings or in audio meetings through the company's communication platform, Lync. The interview sessions, ranging from 20 minutes to one hour, were all recorded and verbatim transcripts were produced. Notes were taken during the shadowing of team members and non-participant observation of daily scrum meetings, knowledge transfer meetings, and informal conversations in the open-plan office in Denmark. Documentary evidence complemented the primary data and consisted of various company artefacts, such as sourcing governance framework descriptions, strategy documents, newsletters, business case descriptions, sprint backlogs (a detailed list of the tasks that the offshore team must complete in a sprint), and internal email correspondences. In total, we spent around 200 hours

in the field and our data sources collected in Denmark and India cover 21 interviews, 59 hours of observations, and a field diary (cf. Table 1). Some interviews were held online, and several hours of observation entailed online meetings, but these were observed from one of the field sites.

Data Analysis

The data analysis started while data were still being collected. In trying to understand the complex whole and the challenges of the work of bridging the new development with the legacy code within GSD practices, our method of analysis was based on an ongoing iterative process of moving between maintaining a close focus on particular situations in the project and keeping a broad perspective of the whole that these situations create [31]. Our aim was to understand – from a close range and a broader perspective – the processes and themes within multiple situations and events in order to illuminate particular challenges within software development in a global setting that involves a legacy system for Danish governmental IT systems. The daily field diary maintained throughout reported on events of the day, as well as questions, thoughts, and reflections on our direction for and inquiry of the research. Thoughts and questions that were noted were sometimes later retrieved when talking with people for verification, or further clarification and elaboration.

We applied approaches from Star [46] for interpretation of our case – understanding the systems and structures involved – to explore information infrastructures that are otherwise difficult to identify. In order to understand the complexities of the SickPay system, we therefore had to investigate the infrastructures in which the the technical system was embedded: in this case, other IT systems, rules, regulations, policies, and sociocultural practices within the Danish welfare system. Thus, the people and systems emergent in our empirical fieldwork data were integral to the information infrastructures that organize and are organized by work practice [10]. In particular, we explored how the existing and inherently embedded rules, regulations, policies, and sociocultural practices affected the GSD work in our case study project.

We read through all the data from observational notes, documentary evidence, and interview transcripts to identify issues and topics that related to the challenges of integrating legacy systems within the Danish welfare system, particularly with global partners involved. In our further work with the data, we compiled several write-ups about the project. By sharing initial findings through informal chats and meetings with those involved in the project, we received helpful comments and clarifications. Working with the write-ups has given us a great opportunity to get an overview of the data, which is central for our analysis [48]. The continuous communication with and reactions of the practitioners have not only served as a form of validation of our interpretations [31], but also helped to drive the themes in our analysis. The insights from the empirical study form a basis from which further investigations can consider the implications of developing a software system within a GSD setting that is to partly replace and conform to other complex IT systems and legacy systems. With our findings, we intend to inform practitioners, as well as research through insights and knowledge that can deepen understanding of the complexities of developing – with legacy systems in complex organizational and infrastructural settings – particularly when both personnel teams and information are distributed across continents. In the next section, we will discuss the details of the case study from which our findings emerged.

RESULTS

In order to answer our research question and understand why the development of technical interfaces within the GSD setup failed to succeed, we here present various examples from our empirical data.

Danish Social Security Benefits

Certain fundamental parts of Danish society are the social security system and its associated benefits that allow Danish citizens to receive cash benefits when sick, unemployed, unfit for work, retired, or on maternity/paternity leave. In Denmark, the labour unions traditionally play important roles in negotiating employer benefits, work environments, policies, and regulations. Moreover, members of labour unions are often entitled to receive unemployment benefits if they are dismissed from their job. In addition, if you live in Denmark, pay taxes on your income, and meet a minimum requirement for employment, you are entitled to cash benefits if you are unable to work due to illness or have been injured. This means that an employee can stay at home during sick leave and receive sickness cash benefits as an economic compensation paid by Danish municipalities. When a citizen – either an applicant or potential beneficiary – applies for sickness benefits, municipal administrators handle the responsibilities of deciding an applicant's eligibility for receiving sickness cash benefits and of calculating and disbursing benefits. The current IT system solution for handling the various types of Danish social benefits – from unemployment benefits to sickness cash benefits – connects to several different systems and solutions within the Danish IT systems supporting the social services and welfare systems.

When a caseworker in a Danish municipality is processing and calculating an applicant's case, multiple systems are invoked and used. For example, there are various categories that a beneficiary will be assessed on, such as whether the beneficiary is unemployed and sick, self-employed and sick, or sick with incoming wages. A beneficiary could, for example, be unemployed and the local service providers could seek to get the beneficiary back into the labour market through employment. This particular task is not handled by the municipalities, but is contracted out to and handled by special job centres distributed all over Denmark.

Thus, when an unemployed beneficiary who is associated to a job centre and receives unemployment benefits becomes sick, the job centres and the municipalities must communicate internally. These entities must share the information that that beneficiary is currently not available to enter the job market, nor entitled to receive unemployment benefits but instead may receive sickness cash benefits.

A New System for the Sickness Cash Benefit

A public tender has been announced for the SickPay system calling for bids on solutions to create a better overview of the benefit case and to make processes more efficient and automated. The goal is to consolidate the multiple information systems for handling the sickness cash benefit into one solution. The public tender provider is *Kombit*, an organization owned by the interest group and a member authority of the Danish municipalities. On behalf of local government authorities, Kombit orders and procures IT solutions and handles the tender competition processes.

The level of multiple information infrastructures that the new system must integrate with is exhaustive. On the extensive loads of data and documents, Kombit writes:

> *the [SickPay] system must integrate with various different solutions and data registers. One of the important integrations is NemRefusion [a common platform for reporting of sickness benefits and maternity leave allowance for companies] where virtually all new cases are born. […] In addition to the NemRefusion integration, there will be integrations of the different job centre solutions, the self-service module at borger.dk [a Danish common public internet portal], basic data records [social security number](CPR), [the Central Business Register] (CVR), […], the municipalities' payment and accounting etc.* (Translated from Danish)

As stated in the above function description, the systems supporting the Danish social services involve various systems. The systems are interconnected in various ways and through different connections, and all follow some Danish standards. Moreover, all are systems that must change dynamically in relation to changes in the Danish welfare system, politics, and legislation. In other words, this is an information infrastructure highly integrated within and influenced by Danish society and the technological advancement of, for example, citizens' self-service portals and mandatory social security numbers. These information infrastructures constantly change: information infrastructures change over time; laws are regulated, changed, or repealed; and IT systems are upgraded or replaced with new systems. Therefore, when developing new systems to take part in the information infrastructures of the Danish public sector, replacing or integrating into legacy information systems like WBenefit, stable information infrastructures will have to temporarily be disconnected and thus appear unstable until the new information infrastructures are well reconnected.

GSD Tasks: Interface Integration

In order to further understand the complexities encountered in a GSD setup when integrating into legacy systems for the Danish social services system, we focus on the particular types of tasks addressed by the offshore team. Based on the argument that the tasks are more technically suitable to the competences in the team, the types of tasks that the offshore team receives from the POA team are interface integration tasks, as these are tasks less involved with understanding the Danish welfare system and legislation:

> *[The offshore team] has primarily received integration tasks, because these are technical tasks that are easy or not easy but may be better suited to the skills that lie [in the offshore team], in contrast to the Danish legislation, which can be a bit challenging if you sit in Bangalore and do not have the [Danish] background right.* (Program manager, interview, 7/2/2013, Denmark)

Working on integrating the SickPay system into various external and internal systems through technical interfaces is therefore the main portion of the PBIs for the offshore team. According to the program manager, the technical tasks that he believed to be "*better suited"* for offshore are also the tasks that are less culturally specific. However, this statement prompted us to delve deeper into investigating *what is meant by tasks better suited for offshore.* While the interface integration tasks may not be heavily grounded in knowledge of and the ability to model the Danish welfare system laws, we saw how these tasks are nonetheless highly interdependent on other internal MData systems, as well as external third-party institutions and systems. After five months with the project, there was a realization of the lack of success with the implementation of interfaces. In regards to the type of tasks assigned to the offshore team, a senior project manager in the SickPay project had the following view on this matter:

> *They [the POA team] have given them the hardest tasks of them all, tasks with the most integration to others [institutions and systems]; where we do not have knowledge in the project, and where we need to establish a lot of appointments with WBenefit. With third party and others, you absolutely cannot be offshore while solving these tasks, so I quite simply do not understand it.* (Senior project manager, interview, 12/12/2013, Denmark)

Interestingly, according to a senior project manager, the interface integration tasks that the offshore team had been given could be among the most difficult ones in the project. However, we would argue that it should be of no surprise that there is a broad range of interfaces that the SickPay system must integrate with: The tender provider already stated this. When the senior project manager raised the

issues with the assigned GSD tasks we wondered why these were not brought into focus earlier on in the project. The challenges described were that the nature of the system is deeply rooted in the Danish welfare system and thus highly interconnected to various other systems, standards, bureaucratic forms, organizations, and people.

Law-Based Systems and Legacy Systems

To find answers for why the technical interface integration tasks in the SickPay project are described by the senior project manager as the most difficult tasks to solve in a GSD setup, we closely observed some of the systems that the SickPay system must interface with. The WBenefit system that Kombit wants to phase out and eventually replace is also developed and maintained by MData and is one of several systems in MData's law-based social services product portfolio. These law-based systems are part of a greater infrastructure of related systems that together or individually serve different purposes in the Danish welfare systems; however, all conform to Danish society, law, and regulation. The law-based information systems in MData are closely connected to other social services information infrastructures that exist both internally in MData (e.g., MData facilitation systems for handling social services such as *cases*, *pension,* and *income*) and in external systems or institutions (e.g., *The National Labour Market Authority*, *Digital Post, job centres, Statistics Denmark*, *Danish Tax Authorisation*, and *Danish Agency for Governmental Management*). This means that data flows through various systems. For example, the SickPay system must collect data from a range of institutions and their systems in the Danish public sector and welfare system. Thus, the system is required to connect and interact with various interfaces.

The WBenefit team has experience with various peculiarities and inefficiencies within the systems that have become commonplace. To ensure the SickPay project did not overlook certain instances of exceptional cases that could cause grave errors for the beneficiary or the municipality, several clarification meetings were held between, for instance, MData business analysts from the SickPay offshore team and the WBenefit team. To progress on a GSD task, the business analyst located at MData in Denmark must collect information from different systems, people, and organizations in order to produce various business refinement documents. These are then handed over to offshore IT developers in India, enabling them to proceed with the development of the task.

When observing several clarification meetings between the MData business analysts from the SickPay offshore team and members of the WBenefit team, the challenges of developing a system to replace some functionality in a legacy system like WBenefit became evident. For example, in relation to assigning a descriptive case number, explaining how a beneficiary's pay period is calculated in the event of their taking a vacation proved to be more complex than just exchanging technically detailed information. Apparently, WBenefit and the SickPay system had two distinct ways of solving the representation of a beneficiary's case number. In the example provided, a senior developer from the WBenefit team explained how the WBenefit team modelled and handled a sickness beneficiary's case number in regards to the law:

> *One thing is the world and another thing is the model of the world, in this case the sickness benefit. […] Say I am sick and I have received sickness benefit, now I go on vacation, and I will then receive benefits from another instance, then I am no longer in the sickness benefit arrangement. [...] This means the case number no longer exists; however, what happens in the modelled world to make it programmable is that the case number is still in the system, and when the person returns to be receiving sickness benefits, then there exists some associations to the "old" case number.* (Observation notes, 7/3/13, Denmark)

This specific example demonstrates how the legislation of the Danish welfare system is entangled with the technical implementation and representation of case numbers in the system infrastructure. The rule is: if you are sick and go on vacation, you are basically not allowed to receive sickness benefits. However, the choice of whether or not to keep associations to old case numbers in the system is an implementation decision made by the WBenefit team. As the business analysts from SickPay realize that they have modelled the case numbers differently, it becomes clear that there is a legacy issue in solving the interplay between the system functionality (legislation) and the system implementation in WBenefit. The challenge is to decide how to handle case numbers when communicating with WBenefit:

> *While WBenefit changed the world in the model, to ease the work, the business analysts [from the SickPay system] and the WBenefit team are now discussing whether this makes any sense, and how they should approach the handling of case numbers. In the end they all agree that the SickPay team should discuss it with the people in the MaternityPay project.* (Observation notes, 7/3/13, Denmark)

While the original idea with this clarification meeting was to seek answers from the WBenefit team, additional questions and issues emerged as entanglements in the information infrastructure of the SickPay system became evident. The differing uses of case numbers also added an additional wrinkle to the MaternityPay project, and created an issue the business analysts had to follow up on for supplementary clarification. This situation demonstrated the chain of dependencies is not only technical in nature, but also entails issues of a more sociocultural character, primarily that the business analysts from the SickPay team would have to perform additional work of a more detective

nature, which was unforeseen and invisible to anyone before the meeting. Thus, the business analyst had to dig deeper inside the MData project portfolio and inquire with members working on yet another project to track down the relevant people from MaternityPay in order to finish MData's part of the work. Meanwhile, the offshore Indian developers, working on building and integrating the SickPay system's interfaces to other systems inside and outside MData, are highly dependent on the work of their fellow team members in Denmark. The MData business analysts in the SickPay offshore team depend on a range of different people and information infrastructures outside of the SickPay project. For the tasks being outsourced, the meeting between WBenefit and the business analysts demonstrated the difficulties or even impossibilities of being able to specify the integration task all at once. While collecting knowledge on a given task, the need for acquiring additional information from additional resources/people became pertinent and required further excavation work.

Building an AS-IS System

In investigating the challenges of developing technical interfaces for the social services system in Denmark, we demonstrated aspects of the role that the WBenefit system and the embedded information infrastructures play. However, and maybe more importantly, one of the primary obligations that the SickPay system must comply with is that the SickPay system is an "AS-IS system," which means the SickPay system should be "AS-the-WBenefit-IS" in its functionality. When, for example, the WBenefit system disburses money to beneficiaries, the WBenefit system is obligated to deliver specific reports (data extracts) to different stakeholders. These stakeholders include members of external systems, the public sector, and institutions involved in the mechanisms of handling the Danish welfare system, for example: *The National Labour Market Authority*, *Statistics Denmark*, *Danish Tax Authorisation*, *Danish Agency for Governmental Management*, etc. In other words, under an AS-IS system, MData – and now the future SickPay system – are obligated via WBenefit to deliver reports to several institutions in the Danish IT system landscape.

Observing the daily work of the one and only IT developer in the offshore team located in Denmark gave us a great understanding of how various information infrastructures were embedded in the SickPay system. The senior IT developer had previously worked on similar projects in MData, thus he was able to draw on his experiences – in particular, the MaternityPay project that is developing a maternity pay system also highly connected to the old WBenefit system. In discussions, he offered insight into the information infrastructural complexities of developing for a legacy system within the social services sector:

> *In the old days they [the public sector/institutions] got the reports from WBenefit, but in the new solution, they need something from WBenefit and from SickPay, to make the world simpler! [laughs] This was also the case in MaternityPay, only that we just made one report and then WBenefit would merge it and send it out. Now I have 98 reports and then WBenefit said, "I will not merge 98 reports, that we are not able to!"* (Observation, 7/29/13, Denmark)

The senior IT developer had previous experience implementing support for reports (data extracts). However, for the SickPay system, compared to the MaternityPay system, the number of reports that have to be created is more extensive and complex due to distinct rules and regulations for handling maternity leave versus sickness benefits. While sickness benefit is handled locally in each of the 98 municipalities in Denmark, maternity leave is handled at a centralized level:

> *[…] I have agreed with WBenefit that I should send an assembled report, so we have to merge these 98 results before we send it to WBenefit, then they only get one report, and from their side it looks the same as the one they receive from MaternityPay – they get one report they should merge. So we have more PBIs in this sprint that deal with this merge concept.* (Observation, 7/29/13, Denmark)

The implementation then must adhere to the restrictions received from WBenefit. The SickPay system must therefore conform to the WBenefit system when taking over some of WBenefit's functionality, including: gathering data from different systems connecting through a range of interfaces, processing the data, and finally handing over one assembled report to WBenefit. The example of creating reports with data extracts demonstrates how conforming to existing information infrastructures is not a trivial task. First, although there are some similarities between implementing social services benefits for maternity leave and for sickness benefits, the legislation and organizational practices of handling the two social benefits are distinct, as they are embedded in different information infrastructures (sickness and maternity), while being connected through their common connection to the same information infrastructure (Danish Government IT). The different information infrastructures impose different implementation approaches that are still interconnected. Second, the precise way that WBenefit wishes to receive data, due to existing information infrastructures, determines and impacts the software development tasks required to finalize the project. In this case, this means the merge concept. Although the technical tasks are assumed to be more suitable for GSD work compared with tasks requiring more heavy business knowledge, the task of implementing reports is not trivial.

As mentioned earlier, the SickPay project depends on both internal departments in MData and external (third party) organizations in terms of acquiring and collecting sufficient information to develop given interfaces and to integrate

with other systems. Due to the many rules in the Danish welfare system, there are various complexities in addition to understanding the business logic and domain when, for example, developing the interface connections for the SickPay system. If a Danish citizen is sick and without job, there are several institutions and systems that need to be connected to the SickPay system. Thus, the SickPay project depends on other institutions, organizations, and system providers to collaborate with them. In particular, during our observations we witnessed how the need to invoke a certain interface (DFDG interface) at a central system at the National Labour Market Authority gave rise to PBIs becoming difficult to complete work on. At first, the National Labour Market Authority was not able to point the offshore team members of the SickPay project in the right direction in terms of whom they should receive information from. In an interview with an IT developer in the offshore team, the effects on the development work were expressed as follows:

> *[…] A part from WBenefit team we faced a huge delay in DFDG interface. […] I have done full development, initially they have given us... three times I have changed, three to four times I have changed the design.* (Offshore IT developer, interview, 11/29/13, India)

As long as there were uncertainties related to how to access a given interface, these were echoed and reflected in the code design that the IT developer was working on. Yet another challenge emerged when the SickPay project finally managed to track down a third-party company that maintained the central system and had knowledge of the DFDG interface:

> *That DFDG started almost in sprint four or sprint five and went on till sprint twelve. From sprint twelve we got some feedback from them.* (Interview, 11/29/13, India)

As we see in the above quote, the tasks related to the DFDG interface were very difficult to work on – not technically, but due to the inaccessibility of acquiring the right information about the interface. Therefore, work was repeatedly postponed over the course of seven sprints, which resulted in the integration with the DFDG interface requiring an additional seven to eight months of work. The people in the third-party company were very busy and no actual agreement existed about obligations to provide these types of informational instructions. Hence, the company was not particularly interested in spending time informing others about the interface. The DFDG interface obviously has a different meaning or purpose depending on whether a person is part of the company maintaining the interface or part of the SickPay project group seeking knowledge of standards to be followed in order plug into and integrate with the DFDG interface.

Security Information Infrastructures

When investigating what it is that makes it difficult to develop technical interfaces for a social services system in a GSD setup, the role and complexity of additional information infrastructures becomes clear. Due to the nature of the systems involved and their embedded information infrastructures, the SickPay project was subject to follow various standards and security rules in MData, as well as rules of the Danish Act on Processing of Personal Data. This meant that some systems and their data could only be accessed when physically located within the European Union. Moreover, in order to access, develop and integrate, and test other system interfaces or web-services, certain security certificates could be required. In the work of integrating SickPay with another system in MData's own social services portfolio, both a test and development authorization were required. We saw that acquiring such authorization prolonged the development work on the PBIs assigned to the offshore team. This became relevant when a developer from India located onsite in Denmark was to work on an interface and process integration task together with some of the offshore colleagues in India. There were several administrative steps to be followed in order to access the different interfaces that the developing system should integrate with. We observed and followed the process of, first, the discussions of the certificate required at the daily scrum meetings for the offshore team, to the eventual ordering of the certificate, followed by a period of a couple of weeks involving people collaborating across teams.

In this case, Michael, a Danish business analyst on the offshore team located in Denmark, was to order a certificate for Parveen, an Indian IT developer temporarily working onsite in Denmark. As the acquisition process could require Danish language skills when talking to security providers or invoking an internal MData portal, Michael worked on the certificate acquisition for Parveen. It was an urgent matter for the offshore team, and even though it was not the business analyst's area of expertise, he took on the task since no one else had thus far. During the daily scrum meetings, it was reported that the first attempt to order the certificate had, unfortunately, not been done correctly. This incidental delay is one of several that occurred in the following weeks, such as key people that could provide help becoming unavailable due to vacations or illness. The type of certificate required was unfamiliar to the SickPay project and, therefore, took more time to setup. The incidents that slowed down the process connect back to the complexities of the information infrastructures involved, including: obligations to adhere to Danish law, the issuing of authorization certificates by MData, a lack of available staff with knowledge on how to work with the certificates, etc. These connected factors all affect the developers' abilities to start their work on the PBIs. Further, while the certificates were slowly acquired and installed, allowing the

IT developers to access the interface, new issues emerged and another person in the project was now being included:

> *There are some problems on some of the PBIs that others in the team are waiting on me to provide some answers on… for example, there is an interface where we use some certificates, and for some reason the certificate is not working on our system testing, but it works well on our development, and why? Well I don't know […] the problem is that I cannot say if it will take five minutes or the rest of the day [to solve]* (Observation, 7/29/13, Denmark)

The issues with the certificate were a recurring topic during many of the daily scrum meetings throughout the five-week sprint and resulted in some pending tasks, or "leftovers," that were added on to the following sprint. A challenge to GSD work practice was the embeddedness in information infrastructure that requires locality (meaning physical presence within Denmark or the EU) to acquire a certificate and ensure the data registration laws were followed.

In the previous sections, we have demonstrated some of the challenges that exist with work in a GSD setup in regards to technical interface integration tasks. In the case of GSD, these types of tasks have been initially preferred over development tasks that require more in-depth business knowledge. But we have demonstrated how these technical tasks are inseparable from work that is more sociocultural in character. Not only can we show that system development and integration with legacy systems are socio-technical tasks, but also that the nature of such tasks and their sociocultural embeddedness – in our case, into the whole Danish social service system and information infrastructure – makes them even harder to be completed as part of GSD development. These tasks could include plugging into different systems through technical interfaces and collecting data about the given interfaces from various systems (e.g., legacy systems, internal, external from third party, etc.), or detecting and tracking down the right information, people, systems, and organizations in order to know how to interface with different systems, following security standards and other requests from relational legacy systems.

DISCUSSION

Through our empirical data we investigated what made it so difficult to develop technical interfaces in an outsourcing collaborative setup of GSD. We set out to investigate the characteristics of software development tasks in order to understand how the replacement of and integration with legacy systems increased the complexity of solving related interface tasks.

When viewing the system interface as an information infrastructure, our data clearly demonstrate the ways in which it would have been impossible to create in advance comprehensive requirement specifications for the task being outsourced. Due to the embeddedness, transparency, and incremental development [46] of the system interface, it becomes clear that no such requirement specification could be made in advance. All these factors existed prior to the development of a project relating the interface to a long list of systems and standards created over time without any pre-determined "person in charge." The very task of "specifying" is an unknown entity prior to getting the work done, since the information infrastructure only becomes visible through its breakdown in practice [20, 23]. The idea that you can send a piece or a module of the larger system to be coded elsewhere rests on the assumption that such clear-cut pieces of tasks exist, and that there exists a presumable coherence within IT governmental systems. It presumes a uniformity of the whole IT system that simply does not exist.

The information infrastructure perspective revealed certain complex aspects of legacy systems. We know from earlier work that, while information infrastructures are continuously changing, there is always a certain stable part, or *installed base* [10], that makes the core spine of the information infrastructure. In our case, we can view the legacy system as the "stable" part of the information infrastructure that makes up the "whole" Danish governmental information infrastructure. The legacy system as an information infrastructure is thus embedded into a range of other structures: all existing laws, legislations, policies, together with the complete set of diverse and interlinked IT systems supporting different practices involved in supporting the Danish flexicurity society. This means that when invoking, developing, or integrating with the legacy system the "stable" part of the system constantly needs to be taken into account. In this perspective, we propose the legacy system as the stable part of the information infrastructure, while the unstable parts are all the connecting systems, related rules and regulations, and the sociocultural practices (and at times invisible work) of finding and acquiring the information required to make work flow.

We join others in arguing that it is critical to look across "multiple levels of granularity, various facets of social life, and diverse technological actors" in order to reveal the stories of (cyber)infrastructure [41, p. 241]. For instance, this could imply that all the tweaking of other systems to function together with the legacy system has to be taken into account when replacing the legacy system. Thus, over time, multiple add-ons, new systems, changes in the policies, etc. have to be designed for and appropriated within the legacy system's capabilities. Our contribution in terms of information infrastructures in CSCW points to how legacy systems can be perceived as stable parts of information infrastructures. We argue that the practices of developing, replacing, or integrating new software within legacy systems have to carefully reckon with issues related to dealing with the inertia of the installed base [47] provided by the legacy system. Over time, subtle design

and obsolete technology are not necessarily capable of explicitly articulating what those issues are.

At the beginning of the paper, we argued that our focus shares some of the same characteristics as the growing body of research on scientific infrastructures; however, our results point to how these characteristics are highlighted differently. The challenges of designing an infrastructure [40], or growing one [49], for that matter, may have little to do with dealing with the development of sub-components and interfaces that are to conform with the inertia of a 30-year-old legacy system. Thus, in our case the uniqueness of an information infrastructure within governmental IT systems differs from the scientific infrastructures in the way that scientific infrastructures are first and foremost built with the goal supporting cross disciplinary, institutional, and organizational collaboration to leverage scientific research [41]. Governmental information infrastructures were not (necessarily) built on the same premises – facilitating multidimensional collaboration and knowledge sharing. Rather, it is a product and a result of a long history of sub-systems coming to live within the context of supporting the continuously changing Danish welfare system, wherein different stakeholders have different needs and approaches toward building the infrastructure. For instance, the government wants to support different mechanisms in the welfare system based on the current government's policies while saving money on expensive software systems, processes, and labour; the software vendors building these systems are interested in offering the most competitive price to win the tender contracts and position themselves within a continuous flow of new system development jobs; and the caseworker at the municipalities would like to be able to process the case handling in a smooth and timely manner. While we do not argue that building a scientific infrastructure can entail several contradictory agendas, we believe that there is a fundamental difference between the two fields. Scientific infrastructures could be assumed to have a common end-goal in their development and construction; however, the challenges of the already existing information infrastructures in the public sector appear to require a bottom-up analytical approach when trying to understand the requirements, source code, and interlinked systems that have sprouted for many years.

Governmental systems are patchworks, in the same way as hospital systems [20], dynamically put together and evolved over time. The homogeneity and tendency toward centralization and standardization of a unifying governmental system does not exist in practice. Rather, the overall system is patched together by the multiple singular systems that have been developed over time. This means that part of working on the task requires the work of tracing back to information and finding out which information is required to solve the technical task. This work is different from refactoring or reverse engineering [33]. Tracing back to information is about asking questions; however, to ask a question you first need to know which questions to ask. The questions that emerged in our case were both related to technical specifications (such as testing edge-cases) and political specifications; however, in practice, it was not easy for the IT developer in India to determine the differences between these, or even to know if they needed to ask questions on this in the first place. Issues that appeared as technical on the surface could emerge as policy-related in practice, and vice versa. For example, how would the IT developer located in India know to ask – yet even contemplate asking – about how the SickPay interface should be related technically to system modules handling vacation registrations, without knowing that in Danish society people can take vacations during their sick leave.

Understanding governmental IT systems requires reflections on and engagement with the social structures within which they are to be embedded [17]. Often, issues arise in miscommunication caused by the difficulties in interpreting requirement specifications in GSD [29]. However, for legacy systems, the issue is even more complex. In these cases, technical issues and policy issues are closely related and embedded in the information infrastructure of governmental systems and institutions in ways that are completely invisible for people who do not work with such systems on a daily basis, or who are only knowledgeable in the business domain and know less about the technicalities.

Part of the task of replacing legacy systems involves identifying the issues and problems relevant for the development of a new system. This must be done in such a way that allows developers to trace back to information by identifying who has institutional knowledge, where those people are located, and if what they know is relevant for the task. In our case, the delay in tracing to knowledge was not about MData not being prepared in advance for the task. Even though the system interface had been developed on its own by IT developers located in Denmark, the developers would still be required to trace back to information as part of their work. The main issue is not about understanding legacy systems, but that no one has all the required knowledge to solve the problem prior to working on a project. No such coherent and uniform whole exists when we are dealing with legacy systems. It is only through *working in practice* on the task that problems and issues are identified in such a way that developers start gaining the relevant knowledge. Therefore, the accessibility and availability of technical and policy knowledge are critical when designing, developing, and implementing legacy system interfaces.

Our data demonstrate how certain kinds of detective work or excavation work are necessary to locate and acquire the right information about parts of the system when developing system interfaces that are technical and also highly embedded in social practices, third-party/external systems, welfare policies, laws, and regulations. Here it is

important to note that excavation work is required because no one knows about all the tweaks made on the legacy system over time. Consequently, there is no such thing as a complete requirement specification existing prior to the project. Thus, identifying or rather excavating the requirement is part of the work of design, since it is only in the very process of designing and constructing the replaced programing code that the excavation work becomes apparent and difficulties arise. In our case, excavation work proved to be highly challenging when business analysts in one location were required to make sense of the system for IT developers located elsewhere. Since excavation work requires both technical and business knowledge, as well as the ability to identify who has the required knowledge for further excavation, one could question the distributed roles of having primarily IT developers in one place and business analysts in another. When managing legacy system replacement projects in GSD, an effort could be made to better balance the technical and business personnel in all locations of the collaborative arrangement, as well as to strengthen the closely coupled work that is required among collaborators [28]. At least in our case, we saw how the business analysts were at times not quite clear on what information was necessary to acquire – for example, on what abstraction level or technical level information was required for the IT developers to start their work.

Part of the work of developing technical system interfaces for legacy systems involves activities of knowledge creation through tracing to, localizing, negotiating, and collecting relevant knowledge. In other words, to obtain the critical information that the developers in our case were bound to participate and interact with involved the whole information infrastructure – including the social organization of work of those who are or have been part of developing the patchwork system (e.g., the WBenefit team, or the DFDG interface provider). These participants could be found within different organizations (since several IT companies in Denmark were developing parts of the patchwork system), as well as the government institutions and systems involved in administrating social services available in Danish society. The work task of developing a technical system interface evolved into a highly complex task revealing an extremely multifaceted information infrastructure without any clear access point for comprehension – a subcomponent of the information infrastructure that could not be logically deconstructed into parts to be sent offshore. Several essential pieces of the task of tracing to relations and associations across institutions, organizations, policies, and technical specifications were opaquely significant, and it was up to the developer to determine which of the connections to follow and which to ignore.

CONCLUSION

Initially, we asked what were the characteristics of the software task of replacing technical legacy systems that increase the complexity in solving the task in an outsourcing collaborative setup. Analyzing our empirical data, we found the framework of information infrastructure very useful in describing and comprehending the complex and intertwined work involved in designing, developing, and implementing governmental IT systems. The conceptual framework of information infrastructures has previously been used to investigate large-scale healthcare information systems and within research aiming toward developing large, distributed, and long-term information infrastructures to support scientific research activities. Our case demonstrates that governmental information infrastructures are not uniform, coherent, or centrally organized. Instead, these comprise a patchwork of embedded systems, which come together in practice to create the fundamental structure of Danish society. These structures cannot easily be understood outside this context.

While our contributions are directed toward the CSCW community, our findings also speak to software engineering by emphasizing the need for a broader focus of a more sociocultural character when examining the nature of software development. This is a need that entails a focus beyond examining relationships between technical dependencies of software systems and the task structure of the development work [12].

We join other software engineering scholars in challenging the idea of reduced communication and coordination due to well-defined interfaces. But we argue through our detailed empirical account that replacing legacy systems within governmental setups is *not simply* a technically pre-defined task. It is a utopic ideal to believe in neat and completely pre-defined software tasks that are easily outsourced. Instead, we argue that software tasks require much excavation work – the work of tracing back to information and knowledge concerning both law and technical specifications, as well as how the information infrastructure has dynamically evolved over time. This work is not easily carried out in a GSD setup.

ACKNOWLEDGMENTS

This study was conducted within the "Next Generation Technologies and Processes for Global Software Development," #10-092313 an interdisciplinary and inter-organizational research project that is financially supported by the National Council for Strategic Research, Ministry of Science, Innovation and Higher Education in Denmark. Please visit www.nexgsd.org for further information. We would like to thank all the anonymous volunteers from ITech and MData who generously offered their time and energy for this research. We also would like to thank Paul Dourish, Lily Nguyen, Morgan Ames, Garnet Hertz, Chris Wolf, and Katherine Lo for their contribution in discussing the argument in this paper. Moreover, we would like to thank Katie Pine and Marisa Cohn for their constructive comments after reading earlier drafts of this paper. Finally, we would like to thank the anonymous reviewers for insightful comments and suggestions.

REFERENCES

[1] Abbott, P. Y. and M. R. Jones (2012). Everywhere and nowhere: nearshore software development in the context of globalisation. *Eur J Inf Syst*. 21(5): 529-551.

[2] Avram, G., et al. (2009). Bridging, Patching and Keeping the Work Flowing: Defect Resolution in Distributed Software Development. *Computer Supported Cooperative Work (CSCW)*. 18(5-6): 477-507.

[3] Bennett, K. (1995). Legacy Systems: Coping with Success. *IEEE Softw.* 12(1): 19-23.

[4] Bietz, M., et al. (2010). Synergizing in Cyberinfrastructure Development. *Computer Supported Cooperative Work (CSCW)*. 19(3-4): 245-281.

[5] Bjorn, P., et al. (2014). Global software development in a CSCW perspective. Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing. ACM**:** 301-304.

[6] Bjørn, P. and F. Kensing (2013). Special issue on information infrastructures for healthcare: The global and local relation. *International Journal of Medical Informatics*. 82(5): 281-282.

[7] Boden, A., et al. (2009). Knowledge Management in Distributed Software Development Teams - Does Culture Matter? In *Proc*. *Conference*, IEEE Computer Society: 18-27.

[8] Boden, A., et al. (2007). Coordination Practices in Distributed Software Development of Small Enterprises. In *Proc. ICGSE 2007*. IEEE Computer Society**:** 235-246.

[9] Boden, A., et al. (2014). Articulation spaces: bridging the gap between formal and informal coordination. In *Proc. CSCW 2014*. ACM**:** 1120-1130.

[10] Bowker, G. C. and S. L. Star (1999). Sorting Things Out: Classification and Its Consequences, MIT Press.

[11] Cataldo, M., et al. (2008). Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity. In *Proc.* of the Second ACM-IEEE international symposium on Empirical software engineering and measurement. ACM**:** 2-11.

[12] Cataldo, M., et al. (2006). Identification of coordination requirements: implications for the Design of collaboration and awareness tools. In *Proc. CSCW 2006*. ACM Press (2006). 353-362.

[13] Christensen, L. R. and P. Bjorn (2014). Documentscape: intertextuality, sequentiality, & autonomy at work. In. *Proc. CHI 2014*. ACM Press (2014). 2451-2460.

[14] Coulouris, et al. (2005). Distributed Systems: Concepts and Design (4th Edition) (International Computer Science), Addison-Wesley Longman Publishing Co., Inc.

[15] de Souza, C. B. and D. Redmiles (2009). On The Roles of APIs in the Coordination of Collaborative Software Development. *Computer Supported Cooperative Work (CSCW)*. 18(5-6): 445-475.

[16] de Souza, C. R. B. and D. F. Redmiles (2011). The Awareness Network, To Whom Should I Display My Actions? And, Whose Actions Should I Monitor? *Software Engineering, IEEE Transactions on*. 37(3): 325-340.

[17] Dourish, P. (2007). Seeing like an interface. In *Proc. OZCHI 2007*. ACM Press (2007)**.** 1-8.

[18] Edwards, P. N., et al. (2009). Introduction: an agenda for infrastructure studies. *Journal of the Association for Information Systems*. 10(5): 364-374.

[19] Ellingsen, G., and Munkvold, G. (2007). Infrastructural arrangements for integrated care: Implementing an electronic nursing plan in a psychogeriatric ward. *International journal of integrated care* (7): 1-1.

[20] Ellingsen, G. and E. Monteiro (2003). A Patchwork Planet Integration and Cooperation in Hospitals. *Comput. Supported Coop. Work*. 12(1): 71-95.

[21] Esbensen, M., and Bjørn, P. (2014). Routine and standardization in Global software development. GROUP 2014. ACM.

[22] Grinter, R. E. (2003). Recomposition: Coordinating a Web of SoftwareDependencies. *Comput. Supported Coop. Work*. 12(3): 297-327.

[23] Hanseth, O. and N. Lundberg (2001). Designing Work Oriented Infrastructures. *Comput. Supported Coop. Work*. 10(3-4): 347-372.

[24] Hanseth, O., et al. (1996). Developing Information Infrastructure: The Tension Between Standardization and Flexibility. *Science Technology Human Values*. 21(4): 407-426.

[25] Herbsleb, J. D. (2007). Global Software Engineering: The Future of Socio-technical Coordination. 2007 Future of Software Engineering. IEEE Computer Society**:** 188-198.

[26] Herbsleb, J. D., et al. (2000). Distance, dependencies, and delay in a global collaboration. In *Proc*. *Conference*, ACM: 319-328.

[27] Hossain, E., et al. (2011). Scrum practices in global software development: a research framework. In *Proc*. *Conference*, Springer-Verlag: 88-102.

[28] Jensen, R. E. (2014). Why closely coupled work matters in global software development. GROUP 2014. ACM.

[29] Jensen, R. E. and P. Bjørn (2012). Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds. From Research to Practice in the Design of Cooperative Systems: Results and Open Challenges. J. Dugdale, C. Masclet, M. A. Grasso, J.-F. Boujut and P. Hassanaly. Published, Springer London**:** 123-136.

[30] Khadka, R., et al. (2014). How do professionals perceive legacy systems and software modernization? In *Proc. ICSE 2014*. ACM Press (2014)**.** 36-47.

[31] Klein, H. K. and M. D. Myers (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Q.* 23(1): 67-93.

[32] Krishna, S., et al. (2004). Managing cross-cultural issues in global software outsourcing. *Commun. ACM*. 47(4): 62-66.

[33] Laguna, M. A. and Y. Crespo (2013). A systematic mapping study on software product line evolution: From legacy system reengineering to product line refactoring. *Sci. Comput. Program.* 78(8): 1010-1034.

[34] Luff, P., et al. (2000). Workplace Studies: Recovering Work Practice and Informing System Design, Cambridge University Press.

[35] Matthiesen, S., et al. (2014). "Figure out how to code with the hands of others": recognizing cultural blind spots in global software development. In *Proc. CSCW 2014*. ACM Press (2014)**.** 1107-1119.

[36] Meum, T. (2013). “Lost in translation”: The challenges of seamless integration in nursing practices. *International Journal of Medical Informatics*. 82(5): e200-e208.

[37] Monteiro, E., et al. (2013). From Artefacts to Infrastructures. *Computer Supported Cooperative Work (CSCW)*. 22(4-6): 575-607.

[38] Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Commun. ACM*. 15(12): 1053-1058.

[39] Randall, D., et al. (2010). Fieldwork for Design: Theory and Practice, Springer.

[40] Ribes, D. and T. A. Finholt (2009). The long now of technology infrastructure: articulating tensions in development. *Journal of the Association for Information Systems*. 10(5): 375-398.

[41] Ribes, D. and C. P. Lee (2010). Sociotechnical Studies of Cyberinfrastructure and e-Research: Current Themes and Future Trajectories. *Comput. Supported Coop. Work*. 19(3-4): 231-244.

[42] Schmidt, K. and L. Bannon (1992). Taking CSCW seriously. *Computer Supported Cooperative Work (CSCW)*. 1(1-2): 7-40.

[43] Søderberg, A.-M., et al. (2013). Global Software Development: Commitment, Trust and Cultural Sensitivity in Strategic Partnerships. *Journal of International Management*. 19(4): 347-361.

[44] Sommerville, I. (2007). Software Engineering, Addison-Wesley.

[45] Souza, C. R. B. d., et al. (2004). How a good software practice thwarts collaboration: the multiple roles of APIs in software development. In *Proc. SIGSOFT 2004*. ACM**:** 221-230.

[46] Star, S. L. (1999). The ethnography of infrastructure. *American behavioral scientist*. 43(3): 377-391.

[47] Star, S. L. and K. Ruhleder (1994). Steps towards an ecology of infrastructure: complex problems in design and access for large-scale collaborative systems. In *Proc. CSCW 1994*. ACM Press (1994)**.** 253-264.

[48] Walsham, G. (1995). Interpretive case studies in IS research: nature and method. *European Journal of Information Systems*. 4: 74-81.

[49] Zimmerman, A. and T. A. Finholt (2007). Growing an infrastructure: the role of gateway organizations in cultivating new communities of users. In *Proc. GROUP 2007*. ACM Press (2007)**.** 239-248.

Paper 3

Matthiesen, S. and Bjørn, P. 2017. When Distribution of Tasks and Skills Are Fundamentally Problematic: A Failure Story from Global Software Outsourcing. *PACM on Human-Computer Interaction.* 1, CSCW (Nov. 2017).[9]

[9] Please note that this paper draw on data from Empirical Study no. 2. However, the company names were changed for further protection of the company's anonymity as well as for the blind review process. In this paper MData is referred to as *SITA*, and ITech as *TechSav.*

When Distribution of Tasks and Skills Are Fundamentally Problematic: A Failure Story from Global Software Outsourcing

STINA MATTHIESEN, Human-Centered Computing, University of Copenhagen
PERNILLE BJØRN, Human-Centered Computing, University of Copenhagen

Using ethnographic data, we provide a critical reflection on the discrepancies between the application of agile development principles and the conditions which render these principles effective for global software development work. This reflection is based on the analysis of a failed collaboration within a global software project, which relied heavily on feedback from mundane project tools utilized for everyday coordination and monitoring. Our study reveals that these tools hid serious issues relating to both the *distribution of socio-technical skills* and a *discharge of accountability in task execution*. As a result, markers of complex collaborative problems were concealed. Furthermore, the imbalance evident in outsourcing setups, which is enacted through high and low status task distribution among partners, further compounds collaboration problems by emphasizing assumptions about remote workers in the absence of direct forms of knowledge interchange.

CCS Concepts: • Human-centered computing → Collaborative and social computing → Collaborative and social computing design and evaluation methods → **Ethnographic studies**

KEYWORDS

Global software development (GSD), distributed work, failure, categories, invisible work, task accountability, distribution of socio-technical expertise; ethnography

ACM Reference format:

Stina Matthiesen, Pernille Bjørn. 2017. When Distribution of Tasks and Skills Are Fundamentally Problematic: A Failure Story from Global Software Outsourcing. PACM on Human-Computer Interaction, *Vol. 1, No. CSCW*, Article 74 (November 2017), 16 pages. https://doi.org/10.1145/3139336

1 INTRODUCTION

Global software development (GSD), particularly as a domain for Computer Supported Cooperative Work (CSCW) research, has provided researchers with many examples of the inherent challenges of interacting and collaborating across geographically dispersed environments. As research has shown, the articulation work [10, 11, 41] required to handle and solve these distributed software development tasks is highly demanding. For example, the task of translating technical specifications across spatial and temporal distance can be prone to failure if common ground is lacking [26]. GSD can also be influenced by socio-politics, such

https://doi.org/10.1145/3139336

as status and power dynamics, issues between local and remote teams, or difficulties at the inter-organizational level that influence a willingness to cooperate [27, 28, 35, 42]. Moreover, people working in GSD projects must rely upon multiple heterogeneous standardized systems, tools, and applications to support their daily work. Many of these systems, tools, and applications are not fully integrated and compatible [43], creating spillover challenges for communicating [15, 24], coordinating [12], and organizing the software development process [1, 23, 40]. Even simple issues like tracing an individual's responsibility within a specific project can often be problematic [52].

Within this complicated collaboration scenario, research has shown that a range of standardized office tools and artifacts, such as slide presentations, text documents, and spreadsheets serve as important tools for supporting collaborative work by reducing the efforts of articulation work [31]. For example, spreadsheets have been found to function as an important tool for requirements change management, when used to summarize and communicate between distributed stakeholders and the team of core developers [31].We refer to this class of tools as "mundane" tools and note their importance in shaping collaborative work [32]. In particular, we explore the role that mundane tools play within collaborative GSD work arrangements. Over the last five years, we have been studying GSD in different organizations. While it is difficult to assess success or failure in software projects definitively [4], one case in our research stood out as a failure because it failed to facilitate any collaborative processes across geography. This resulted in the code, produced by the developers located in India, never being utilized. Given that only a small number of failed GSD projects are reported in CSCW research [12], we saw this as an opportunity to understand what went wrong—especially with regard to how mundane tools made certain parts of the work visible and other parts invisible.

In this paper, we report from an ethnographic study of a GSD project between an Indian IT vendor and a Danish IT company. We explore the invisible work [55, 57] that made the main organization (i.e., the Danish company) not notice problems arising in the project in due time. Investigating the role of mundane tools, we reveal two important problems that the project encountered, namely the challenges related to the distribution of socio-technical skills and discharge of accountability in task execution. Both of these problems were missing in the formal representations displayed in tools and practices in the project, and thus received no attention in daily interaction, coordination, or communication. Our data show how the inappropriate task distribution, combined with problematic distribution of socio-technical skill and expertise across sites, was lacking formal representation and, thus, was not included in important decisions. The underlying classification scheme embedded in the monitoring tool and the organization of the daily collaborative practices provided limited visibility of fundamental problems, which resulted in the markers of failure remaining undiscovered until it was too late.

This paper offers a two-fold contribution: first, we provide nuanced and multi-faceted empirical data that details a failed case of GSD. Such data is often difficult to gather as it requires a high level of access to organizational decision making, which can be hard to obtain in most companies. Second, we demonstrate how mundane tools accentuate aspects of velocity and black-box fundamental problems in the GSD setup, which combine to offer little margin for communicating issues encountered in the distributed collaboration. We do not argue that the project failed because of the mundane tools. Instead, we argue that when organizations utilize mundane tools (such as Scrum backlogs and Burn-down charts from agile software development frameworks [53]) as their main tool for monitoring performance indicators, there is a risk of occluding vital markers or voices. It is this occlusion that may lead to a derailment of the collaboration.

2 RELATED WORK

Failure of software projects is not exceptional [14, 20, 63], and tools and methods decreasing software risks are considered core for enhancing project performance [3, 25]. Factors which can increase the risk of project failure may include various dimensions of a project such as team members' lack of specialized skills, constant requirement changes, technical complexities, corporate politics, user resistance, or poor project planning [25]. Projects can also fail due to weak business cases, a lack of involvement by top management [63], as well as subversive behavior by stakeholders, which can sabotage any software project [51]. In a global setting, subversive behavior might occur in situations in which local engineers feel threatened when

asked to educate and train less expensive, foreign versions of 'themselves'. Clearly, social, political, hierarchical, and organizational issues have implications for the success or failure of distributed work [2], including team dynamic and power [27], communications and conflicts [28, 29], coordination of expertise [21], domain knowledge [36], and language-related inefficiencies [46]. Furthermore, as risks can also include various stakeholders across all of these various dimensions—a reality that reflects the complex multi-dimensional nature of any software project—the impact of software risks can vary widely.

As is hinted at above, many important aspects of the work related to global software development are not formally articulated; they are implicit or unspoken. Work, as we know it, is never truly invisible or visible [57]. Instead, it is the practices, meanings, and work structures that emerge in a cooperative work situation, which constitutes the work itself [30]. In our effort to understand the reasons why management did not notice or acknowledge the problems that arose in our case, we focus on the hidden or taken-for-granted activities in the project. In exploring this invisible work – the implicit articulation work – we look beyond formal descriptions and pay attention to unrewarded aspects of GSD work, which is often left out of formal task descriptions but is nevertheless critical for executing a project successfully.

In CSCW research, studies of invisible work [9, 55, 58] have proved seminal in amplifying our understanding of professional work and the connections between the subtle features of work. For example, in the medical profession the role of nurses has been 'discovered' as important and thus has been reintroduced into the design of hospital systems [7]. Making nurses' work visible was vitally important for moving the medical profession to include the multiple interdisciplinary areas of medical work into their schemas [13, 62]. In this way, invisible work has proven useful in giving a voice to the performance of work that is not acknowledged or even noticed by others [22, 44, 55, 57, 61]. Focusing on work that otherwise goes unnoticed enables legitimacy and thus empowers those performing the work [e.g. 57, 59, 61].

However, making work visible in this way also introduces new agendas of control and surveillance once the work is open to scrutiny and monitoring [57, 59]. This scrutiny often takes place via collaborative technologies, which embed the ways in which work is formally valued and categorized into their designs. In other words, the design of a technology can influence who is deemed to be performing the 'high' status work and who performs 'low' status work. What we notice or neglect in collaborative work is fundamentally shaped by the underlying assumptions and the available classification schemes that are embedded into the technologies or the practices applied [56, 57].

Technologies also reflect and perpetuate *unconscious* cultural assumptions such as values, attitudes, and worldviews [54]. These unconscious opinions shape the opportunities and constraints that people have with technologies [33, 38, 50]. The consequences of such categorizations and calculations become particularly salient when work is distributed, an argument that is supported in Martin's study of call center work [39]. In our case, exploring why the core collaborative problems in our project were not discovered until it was too late, brings to the fore the need to understand the values, attitudes, and worldviews inculcated in the use of mundane technologies and their associated practices. In particular, we unpack the ways in which the embedded classifications within monitoring tools used in distributed software development risk the black boxing of organizational decision-making. Building on prior research that shows that the heterogeneity of artifacts (e.g. technological tools) in use in an organization reflects its social distribution of skills [48], we focus on how tools can shape what is not visible in a distributed collaboration.

3 RESEARCH STUDY

The authors had the opportunity to conduct several workplace studies [37] and collect ethnographic data [49] in some of the largest IT-companies in Denmark as part of the interdisciplinary research project Next Generation Tools and Processes for Global Software Development (nexgsd.org). The emphasis of this research was on the relationship between software development technologies and collaborative practices in globally distributed work. The larger project also involved another set of outsourcing and offshoring studies in companies in India and in the Philippines [8, 15, 19, 34, 41, 60].

The data presented here is part of a sub-study conducted in a large Danish IT company, here called SITA. SITA has several branches in Denmark and a history of more than 40 years of delivering, developing, and

administering IT for the Danish public sector. In this paper, we draw primarily on the data collected at SITA's headquarters in Denmark, however ethnographic work was also done in India.

Companies in Scandinavia started to engage in globally distributed work by outsourcing software development jobs to lower wage countries such as China or India as far back as 2006. For SITA, India presented an opportunity to recruit highly skilled labor with technical skills that were aligned to the development of enterprise application software. By 2013, SITA employed more than 3000 people in Denmark and engaged 200 people globally at various IT vendors located in Poland and South Asia. An internal department unit named *Global Delivery (GD)* supported and improved outsourcing projects at the company.

When we started our fieldwork at SITA, in 2013, we planned to study multiple projects with global outsourcing components. However we decided to focus on one large scale governmental IT project called 'Rocky' (a pseudonym) which was of particular interest due to the EU mandate for public tender competition which allowed (strongly encouraged) SITA to include outsourcing in order to put together a competitive bid on the development of an IT system for administering welfare benefits in Denmark. This public IT system was intended to optimize and automate the preservation of updated case management data within the Danish social welfare system by establishing a modern, municipal IT support system that included enterprise resource management (ERP) and customer relationship management (CRM) features. In order to achieve this goal, the IT system needed to integrate several public IT system solutions and registers that were subject to governmental policy and legislation. The system also needed to account for the way that governance rules are used in the daily practices of governmental caseworkers.

At the start of our investigation in 2013, the project employed approximately 60 people with various technical and business domain-related competencies who worked from one of three different locations: SITA's headquarters in Denmark, a SITA subdivision location in Denmark, and a large and well-regarded IT company in India (pseudonymously called 'TechSav' here). A team of 13 IT professionals worked from the TechSav location in India, referred to as 'the offshore team'.

We were told that this project was very successful, representing the 'best practice' in GSD. Interestingly, Rocky had a difficult start with several less than successful iterations; however, the Global Delivery department informed us that participants in Rocky had solved the issues and that the project was finally on the right track. Despite the known difficulties in executing large governmental IT systems via outsourcing [40], we oriented our focus on the *organization of work* on the Rocky project rather than the tasks being undertaken. It was with this background that we commenced our data collection.

3.1 Data Collection

Our empirical work was done over a six-month period in 2013, with follow-ups in 2014 and 2016. The focus of data collection was to account for the everyday practices [9, 49] of people working collaboratively in globally distributed situations. We collected qualitative data in India and in Denmark based upon approximately 60 hours of shadowing project participants and observing various collaborative activities (i.e., co-located and virtual meetings). We interviewed 22 people, 15 from SITA and 7 from TechSav, including developers, testers, business analysts, project managers, Global Delivery managers, an end-to-end project leader, a program manager at TechSav, a delivery manager at TechSav, a Global Delivery director, and a general manager at SITA, who was also a steering committee member in Rocky. The first author was assigned a corporate laptop in order to participate in technology-mediated activities (utilizing collaboration tools), analyze documents, and gain access to other project-relevant information such as email and the project's SharePoint document management system.

Our empirical data comprises interview transcriptions, observational notes with thick descriptions, and a fieldwork diary with memos and self-reported statements. The diary includes reflections on empirical challenges and records analytical and methodological choices [5]. We also collected various internal company documents including meeting agendas, product backlogs, sprint plans, sprint backlogs, business refinement documents, interface descriptions, and function descriptions as well as a number of general

internal documents such as global outsourcing strategy memos and an internal survey reporting on employees' cross-cultural collaboration mindsets.

3.2 Data Analysis

Our data analysis began during data collection as we iteratively revisited, as we collected the text materials (observations, interview transcriptions, emails, self-reported statements), photos, screen shots, system documents, team slides, sound files etc. This process enabled us to continuously develop, categorize, and test emerging themes in the data [16]. Once all of the data was collected, we utilized these themes to formally analyze all of the data by producing multiple rounds of rich empirical write-ups [17], in which we zoomed in on exploring the problems experienced in Rocky project and developed our interpretation of the reasons for these. We discovered an important tension between project participants' attempts to flag core problems and their inability to outline their concerns in useful ways to relevant people. We got intrigued with "the invisible" dimension related to the use of mundane tools because we found that tool use on the project created particular types of representations. This insight led us to analytically identify what kind of information was present or absent in the use of the technical tools on the project. We determined that information that was missing from the representation was rendered to residual categories of work and left out of the everyday assessment of progress [54, 59].

By focusing on the residual categories, which can pose political and ethical challenges for the design and use of technologies, our analysis emphasizes categories of work that lack formal representation in the classification scheme [56] that underpins the design of the mundane technologies in use on the project. By using residual categories as an analytical instrument, we are able to identify and investigate the people and work that exists outside of the tools' existing classification scheme(s) [13]. In making this analytical move, what became apparent was the way that certain residual categories related to details, contingencies, and variations of everyday work were not represented [56] in the design.

Accounting for work within residual categories also helped us see that the main challenges in Rocky were not evenly distributed across the different geographical sites; in fact, and to our surprise, the majority of the problems were related to SITA in Denmark. In fact, the imbalance caused emergent consequences (e.g., dissatisfaction of the code deliveries, organizational restructuring and layoffs) that *only surfaced* in Denmark. Problems never reached the developers at TechSav. As a result, the developers in Denmark did not invite the developers in TechSav into the discussion and due to constrained information flow, TechSav employees were neither closely involved with, nor aware of, what was going on in SITA.

This finding prompted us to consider the imbalance as an important opportunity for investigation. The imbalance turned out to be fundamental for why problems in the organizational setup were not discovered in due time. Data analysis directed our attention towards the analytical themes forming the imbalance, which we articulate as the *distribution of socio-technical skills* and the *discharge of accountability in task execution*. The development of these themes, which rely primarily on quotes from the interviews at the main SITA location (given that the TechSav employees knew little to nothing about these concerns), serve as the core analytical contributions of this paper.

4 RESULTS

4.1 Collaborative Tools in Practice

In order to coordinate and monitor the global work in Rocky, participants from TechSav and SITA gathered for daily Scrum meetings using Lync (now called Skype for Business). A central focus of these meetings was the sprint backlog. The sprint backlog is a spreadsheet that provides an overview of particular development tasks, referred to as the product backlog items (PBIs), to be solved over the course of a five-week sprint. The standardized attributes of an individual PBI include an ID number, a description/name (e.g., "Transmit answer about X information to CRM"), and a range of sprint tasks (sub-activities or stages) that are necessary for completing the PBI. Each PBI also contains a sprint task that is categorized as "CRM

development, Unit test & documentation", "Test case preparation", "System test", "Acceptance test", etc., which means that a PBI moves between various areas of professional responsibility in the organization. Each sprint task is attributed to a specific business analyst, CRM developer, etc., who is in charge; it is also assigned an estimate of the hours required to solve the PBI at each respective stage.

During the daily Scrum meetings, the Scrum master goes through the sprint backlog by sharing his screen and running through all the PBIs currently in process. Each participant in the meeting must account for and update (reduce or increase) the time estimate of each PBI by stating the remaining hours of work and, by consequence, the hours that have been "burned down" (spent) on the PBI in the stated time period. Finally, each participant must report whether the PBI was 'done'.

The burn down of PBI hours serves as an important indicator of performance at SITA because it is translated into a measure of velocity, basically representing how fast the different development teams in Rocky can produce PBIs in relation to estimates. Calculating the "burn down" of hours as a measurement for the performance and progress of work is a well-known practice in agile projects, such as Scrum [53]. However, the way it is used varies within organizations and even within individual projects. In the numerous daily technology-mediated Scrum meetings we observed during our fieldwork at SITA, the Scrum masters' main focus was discussing PBI estimates and burn down of hours in the sprint backlog. By comparison, little time was spent informing or even exchanging information about the actual work, such as clarifying specifications or handling unanticipated bottlenecks. One senior developer shared his thoughts on SITA's emphasis of quantitative measures:

> "That is the downside of our way of running Scrum [...] if we can say we have finished our PBI, we are required to say so, and if it is finished with a defect, we are [still] required to say we finished the PBI. So you lose focus on your quality because it is constantly focusing on burn-down PBIs." (Senior developer, 12/16/2013)

As is clear from this comment, reviewing sprint backlogs with a focus on PBI estimates was central to the way performance was assessed and evaluated during SITA's daily Scrum and weekly project meetings. The sprint backlog spreadsheet indicated not only quantitative measures of project performance and speed, but generally assessed how things were going in the sprint overall.

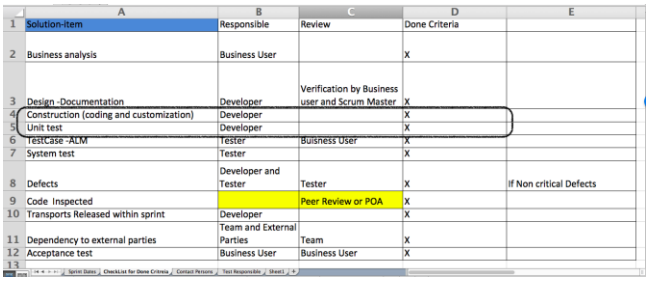
4.2 Cracks in the Surface

Since the basis for monitoring a project was based largely upon measures of "burn down" hours, there was little attention paid to the qualitative aspects of the work. This skewed attention proved critical. Five months into our fieldwork, Rocky entered a state of emergency because of a combination of last minute requirement changes from the customer and changes in the resources allocated to the project. Within SITA, Rocky had been deprioritized and staff were re-allocated to accommodate other projects in the organization. Some employees associated with Rocky risked receiving a redundancy notice at the end of the project as the CEO announced that the company would be laying off 10% of the staff in the coming month. The offshore team was impacted by this news when the Scrum master located in India was downgraded to daily team leader, meaning that he was stripped of any authority.

These changes wrought a period of upheaval and disruption for the remaining employees in Rocky. Closing projects and late change requests from a customer are natural troubles in software development, yet these changes were more surprising for those involved. The project leader in Denmark, Bjarne, who had managed to turn unsuccessful sprints into successful collaborations, had been dismissed. Jens, an external consultant and project leader, who was brought in to steer the software development in Rocky through the final sprints, replaced him. Dennis, a tester who worked closely with the offshore team, explained the awkward situation to us, detailing how a furious executive manager had abased Bjarne in front of the whole project team because he had supposedly presented inaccurate estimations and progressions of Rocky to management. According to the upset executive manager, Bjarne was guilty of disguising the realities of the project.

The coming weeks saw more staff changes as several other IT-developers resigned, including the former Scrum master in India. At the same time, the new management group in Rocky started to review and

evaluate the complete set of deliverables in the project. The quality of the work carried out by the offshore team was assessed to be highly unsatisfactory. The situation devolved further when a divergence between what had been marked as delivered in the product backlog and what had actually been produced was discovered. This divergence arose in a review of what had been categorized as 'done' without actually being 'done'—a situation that had occurred despite everyone's knowledge of SITA's formal definition for project completeness. As shown in the "Checklist for 'Done' Criteria" (Fig 1), the interpretation of 'done' was equivocal as no formal review criteria were stated for the solution items. Neither 'Construction (coding and customization)' nor 'Unit test' stated any formal review mechanism, which may explain part of the reason for the confusion regarding the requirements for a task to be checked as 'done' that existed across sites.



	A	B	C	D	E
1	Solution-item	Responsible	Review	Done Criteria	
2	Business analysis	Business User		X	
3	Design -Documentation	Developer	Verification by Business user and Scrum Master	X	
4	Construction (coding and customization)	Developer		X	
5	Unit test	Developer		X	
6	TestCase -ALM	Tester	Buisness User	X	
7	System test	Tester		X	
8	Defects	Developer and Tester	Tester	X	If Non critical Defects
9	Code Inspected		Peer Review or POA	X	
10	Transports Released within sprint	Developer		X	
11	Dependency to external parties	Team and External Parties	Team	X	
12	Acceptance test	Business User	Business User	X	
13					

Fig. 1. Checklist for 'Done' Criteria in project Rocky

Once the severity of the situation became clear, management realized that the global collaboration was far from the success that they had anticipated. The offshore team was subsequently demoted to a support function and assigned to do ad hoc tests and development tasks outside the project's formal tasks. The offshore team was never informed about, nor confronted with, the fact that SITA was unsatisfied with the offshore deliveries. Instead, the restructurings at TechSav took place without further notification. Indeed, during this entire period, the TechSav people were not fully involved in the changes and the situation in Denmark was not explained to them. This became even more apparent during our fieldwork in India:

> After the weekly project meeting [online] the offshore team started to discuss the meeting. Several of the developers said that they did not understand why people at SITA were talking about not being able to accomplish the PBIs and that they [SITA] were now looking for more ABAP developer resources, when at the same time they [the offshore team] were soon out of work. To them it made no sense, [...] the offshore team was no longer part of any solution team. (12/04/2013, notes, India)

The offshore team in India was more or less left to themselves with the exception of a daily conference call with managers from SITA, which usually lasted about 10 minutes. Observing these meetings from TechSav, the interaction with the offshore team appeared to be highly deprioritized, and the meetings, resembling the daily Scrum meetings in outline, were in reality only loosely structured if compared to the previous meetings we had observed. In various ways, the offshore team became progressively less included in the project collaboration and they were assigned fewer and fewer tasks. Contrary to the work performed in Denmark, the work of the offshore team was no longer mentioned or discussed during the weekly project meetings. When observing Rocky meetings (both from Denmark and India), the offshore team in India participated silently without any concrete role in the project. *"Maybe this is because we are doing very insignificant work,"* an offshore developer said to us during the field visit in India. When Rocky ended a couple of months later (as originally planned), the IT-system had not reached the deployment phase. Indeed it never did so.

4.3 Encountering Tensions

In order to examine potential signs of emergent problems, we return to our analytical focus on the project prior to the dismissal of project leader, Bjarne. Two particular questions linger here: why were issues of quality in the delivered code not discovered earlier and why did such wrongful progression measures persist?

Despite the rhetoric of Rocky as representing 'best practice' in global outsourcing, it turned out that the offshore team's deliverables had been faulty from the project's early stages. However, these problems never surfaced during any of the status meetings. When the SITA IT developers finally pronounced that all of the technical interfaces produced offshore were erroneous and needed re-development, it became clear that the causes of these critical problems did not fit into the progressions displayed in the project meetings. For example, the offshore team's daily Scrum meetings, the weekly Rocky status meetings, the sprint review, and the planning and retrospective meetings all gave the strong impression of successful delivery.

When confronted with the question of why they had not disclosed these issues earlier, the IT developers in Denmark explained that they were not allowed to share their frustrations. One IT developer explained how she experienced that the IT developers in SITA were not allowed to criticize the global outsourcing arrangement:

> "[…] people do not dare talk because global outsourcing has been decided to be a success, and then you cannot say aloud that you think it is difficult." (08/02/2013, notes, Denmark)

The SITA IT developer explained further that during a common project meeting at the headquarters, several people from Rocky had expressed their growing concerns and dissatisfaction about the global collaboration. However, as a result of their articulated concerns, the developers had been taken aside and given a reproof by the (now former) project leader, Bjarne, and his executive. While the IT developers unofficially experienced criticism as undesirable, the official management style at SITA was articulated as encouraging critical voices:

> "[...] here, we do not keep a lid on, so I have also had evaluation meetings, where they [project participants] also got to vent some of their frustrations, and of course there have been some under way, but we have also addressed them all together. […] but, yes, of course there is always room for improvement and things can be done better, and there aren't things we should not say - we'd rather know it, and relate to it instead." (07/02/2013, interview, project leader, Denmark)

This project leader's statement stands in contrast to the experiences shared by the SITA IT developers. To further emphasize their frustrations, one developer even suggested that the project would be better off if they handed over all the offshore tasks to one of the senior SITA IT consultants and paid the whole team in India to do nothing. This statement demonstrates the frustrations of the senior developers, who were required to spend additional, unacknowledged time to support the offshore developers' PBI work, while simultaneously attending to their own PBIs. Measuring senior developers on only the visible part of their work may have also contributed to the tension on the project.

4.4 Confronting the Tensions

Our initial agreement with SITA was to present our empirical research insights to the organization to help them improve their global collaboration. However, we encountered unanticipated tension when we confronted SITA with our findings. Dismissing any claims about the issue with the offshore PBI work and task deliveries, for example, Bjarne, the project leader said:

> "It's not true […] we dealt with it and looked into how much of this was right or wrong. And it's not correct, it's not true [laughs], so…" (08/13/2013, interview, project leader, Denmark)

The tension here, particularly around conflicting reports on the tasks developed offshore, raise a fundamental question: why would a developer claim interfaces to be erroneous and the project leader deny the problem? Our interpretation of this contradiction is that it reflects an understanding of accountability on the project—an issue that was exacerbated by communication and collaboration struggles. While

management might describe accountability contradictions as fundamentally grounded in employees' fears of job insecurity and concomitant resistance towards global work, our data reveal a different explanation. We suggest that the accountability contradictions on the Rocky project have to do with the unacknowledged efforts and the lack of vital technical skills and knowledge required from a range of SITA workers in order to complete the task.

When looked at closely, the (apparently) inaccurate estimations and progression reports submitted by the project leader, as well as his dismissal, revealed conflicting accounts of the quality of the offshore deliveries. Both the monitoring tools and the practices applied during the daily Scrum meetings omitted representations of non-tasks or informal aspects of the work, which were critical for maintaining a well-functioning collaboration across sites. Instead, concerns and concrete problems were glossed over while the manifold constituents that comprised the complexity of the collaboration were discovered too late. Put simply, although one can understand why a project leader might want to represent project progression positively, the question remains: why were there no channels for the IT developers to express their concerns about deliverable quality?

4.5 Enacting Global Collaboration in SITA

Stories of unsatisfactory deliveries serve as an expedient means to undermine the offshore team. But the relationship of these narratives to SITA's organization of global outsourcing and collaboration deserve further attention. For example, business analyst, Anders, told us that Bjarne and the whole offshore team were made out as scapegoats for why Rocky failed; they had been perceived as a lightning rod. Additionally, the lack of success was related to the overall strategy for global work in SITA:

> "Perhaps it is also a little naive to think that we [SITA] can outsource [development work] to a completely different team without any waste in hours. You know, the idea that it doesn't cost anything to move from A to B. Then you are a little naive." (12/16/2013, interview, Senior IT developer, Denmark)

Interestingly, the statement above indicates an inconsistency. The IT system under development by SITA and TechSav was to be part of the Danish public sector meant to enable governmental caseworkers to administer welfare benefits. The Danish welfare system cuts across several public subsystems forming a large information infrastructure that is closely connected to governmental policy and legislation. The system also encodes how governance rules are to be used in a caseworker's daily practices. As a consequence, project development involved important distinctions between technical understanding and business logic. In Denmark, SITA allocated a business specialist, Arife, to clarify business related questions for the offshore team at TechSav. Trained as a caseworker, Arife knew all the relevant legislation and associated procedural requirements related to the caseworkers' daily practices. Hired as a specialist, Arife was responsible for translating and explaining the business requirements into related IT goals. However, as the quote below shows, Arife did not have the necessary technical knowledge to translate the business requirements into the requisite technical form for the developers in India:

> "What happens is that from the very outset I'm alone with 13 people sitting in India, […] I have never heard of SAP systems, and all of the terms they use, I don't know anything about it, and this is the beginning - I'm all alone here [...] and I am supposed to provide clarifications. The first clarifications were about interfaces, where you must sit and define parameters and I have never - I had no idea of what a parameter was!?"(08/13/2013, interview, Business Specialist, Denmark)

Business specialists such as Arife are extremely important for projects such as Rocky that involve important details about legislation and caseworker workflows. Nevertheless, Arife's expertise as a business specialist did not include the ability to translate business requirements into functional requirements, data tables, or parameters. This means that there was a clear—and problematic—semantic gap in the project; no one offshore had the requisite knowledge for translating the business requirements into technical requirements. As a consequence, the offshore team did not have any means or opportunity to obtain the information required to complete their tasks successfully.

4.6 Learning from Failure

Rocky's offshore team was primarily geared towards integrating third-party systems in to the IT system to correspond with the complex Danish governmental IT landscape; they were also responsible for integrating the IT system they were developing with several public authorities in Denmark. In practice, the skills available on their team did not align well with the work required. According to the project leader, Bjarne, they were aware of this issue:

> "The Danish business analysts and specialists were not good enough, and we were aware of that [...] but there are also restrictions on how many resources you get. Well, you can ask for it, but that does not mean you'll always get them and with that there are consequences - that was one of them!" (08/13/2013, interview, project leader, Denmark)

Though SITA has been delivering IT systems to the Danish Government for more than 40 years and has also been engaged in global outsourcing for several years, it could be argued that they were naïve when it came to the collaborative setup between the offshore team and the support mechanisms that were provided onsite. Perhaps this was due to an unfamiliarity with the characteristics of the work sent offshore, but more likely this was result of SITA failing to learn from earlier projects—a fact that was explained to us by several project participants including developers, business analysts, and project managers. Prior to project Rocky, SITA had run another project called Matterhorn that aimed to develop a comparable type of welfare benefit system to that of project Rocky. As it turned out, the development plan had been organized in similar ways, down to the tasks sent to the offshore team in Matterhorn bearing a striking resemblance to the PBIs assigned to the offshore team in Rocky. Matterhorn had failed one year before our study of project Rocky. Reflecting upon the global work arrangement in Matterhorn, an experienced project manager had a bittersweet view of the situation:

> "[...] you even had lessons learned from project Matterhorn where the work failed completely and here they also worked on the same kind of PBIs [starts laughing]! Well, that is simply... that is to shoot yourself in the foot with a bazooka [laughs]. So, yeah it is quite unbelievable that we did not learn from it!" (12/12/13, interview, project manager, Denmark)

According to several project participants, the tasks were extremely complex in nature and required significant elements of business logic combined with highly technical skills. The difficulty in representing tasks such as this for an offshore team may partly explain why the work had to be re-developed in Denmark. Even so, we recognize that the reason for failure on both of these projects has less to do with a lack of technical knowledge and more to do with how that knowledge was distributed; generally speaking, the socio-technical organization that SITA set up was inadequate for the task. For the most part, various assumptions concerning task allocation remained implicit. Hence, as a project manager in SITA later put it, it was his impression that the offshore team had been assigned the PBIs no one else in Denmark wanted:

> "[...] there was simply a situation where a Scrum master and so on [product owner, IT architects etc.] sat down and said: "I don't feel like doing this", "I don't feel like working on this", "I would really like to do this" and "I would absolutely love to work on that, but I don't want to have anything offshore, that I'd rather not touch", "Well, fine, then you get this and you get that and let me see, what do we have left? Well, we have offshore and we have reports, and we also have no Scrum master left ...well ok, let's send this off [to India]"It's terrible!!" (12/12/13, interview, project manager, Denmark)

The reasons for outsourcing the work in the first place might not have been made entirely explicit. It is therefore legitimate to ask if neither the product owner nor the Scrum master in Denmark were interested in using global outsourcing, why try to do so? According to another manager in SITA the decision to outsource is primarily economic—it is *"a question of being able to compete on the hourly prices and that is why you need to use cheap labor from outside of Denmark - otherwise you will not win the contracts."* In order to win and launch a project at all requires a competitive business case. Nevertheless, we see contradictions in how public IT projects are construed in relation to the applied development framework and the application of related tools:

> "We are running a sort of a 'SITA agile project', which means… well it is not entirely agile, as we are operating with a fixed scope we have to deliver [on the project]. But there is a product owner, who is responsible for creating the backlog, and then we run sprints of five weeks" (07/02/2013, interview, project leader, Denmark)

What the project leader explains above is the contradiction of applying agile development approaches to a product that is by nature fixed in due to highly elaborate requirements from the public tender. Seeking to apply an agile development methodology—where the fundamental idea is that the solution and requirements evolve and adapt as the development progresses—to a rigid IT goal requires, at best, a great deal of tailoring to be effective. In our case, the 'SITA agile project' approach basically meant applying agile development roles, tools (sprint and product backlogs, burn-down charts, etc.), and ceremonies (sprint review, daily Scrum meetings) to a project with little room for adaptive changes in plans, continuous development, or agency for fostering collaboration within self-organizing and empowered teams.

Software project failures emerge in the mist of complex, inter-related organizational, financial, social, and semantic issues [e.g. 3, 25], however the specific ways that a failure takes shape, we argue, is closely tied to the organization of *global* software development itself. The organization of the globally distributed setup in Rocky failed at various points. Specifically, it was assumed that the division of labor between the two sites was a purely task-based, not an organizational, matter [36]. In the following section, we discuss these organizational features as we seek to identify the unintended ways that mundane tools shaped the collaborative work in Rocky.

5 DISCUSSION

5.1 Distribution of Socio-Technical Expertise Was Not Recognized

Frequent interaction is important for successful collaboration in globally distributed settings [28, 29, 47]. Closely coupling work across geographical locations by obliging people to engage with the extra articulation has been recognized as an important parameter supporting distributed work [8]. Interestingly, our case involved both frequent interaction and closely coupled work across geographical locations; however, problems were still present. The offshore team could not do their work without close interaction with the onsite team. The project task required different specialized professional competences, such as knowledge of the product, the domain, and connecting technical systems, yet the required expertise and skills were invisibly [55] undervalued. Indeed, GSD work requires interdisciplinary skills to perform the necessary activities [2, 3], but Rocky proved to require technical expertise to a much higher degree than expected. Thus, the role undertaken by the business specialist in Rocky was crucial for bridging domain knowledge with technical knowledge [36, 48]. The issues of flawed code in the technical interfaces did not reflect poor programming skills by the offshore developers, but rather highlighted how poorly the distribution of socio-technical expertise was organized and structured across project sites. Moreover, the translation of domain knowledge and business logic into suitable technical parameters and requirements was a major problem, causing low quality in code deliveries. Ultimately, it became clear that the domain expert allocated in Denmark did not have the expertise required to translate domain insights and technical requirements across contexts and professions. Unfortunately, this cross-functional and socio-technical expertise—so important for supporting the remote developers in succeeding with their development work—was rendered invisible [55,56].

5.2 The Outcome of Meetings Were Assumed to Stand Proxy For the Real Work

Turning to the role of technology, we find that the work structured around tools [30] only presented data about the project in the form of quantifiable measures, primarily relating to the speed (velocity) in which tasks were carried out. Notably, these same mundane tools did not include or support the underlying classification scheme(s) [13, 55] of the associated code reviews; they excluded qualitative parameters on the Rocky project altogether. Generally speaking, software methodologies such as Scrum have been shown to strengthen the structure and organization of software projects [53], however weekly project meetings, including discussions of sprint backlog results, omit non-calculable considerations of global collaboration arrangements. Looking more

closely at the sprint backlog (the primary tool used during these meetings), we see that it renders the manifold qualitative aspects of work as well as considerations of expertise distribution [36, 39] outside of the classification scheme embedded in the tool [56]. These types of organizational complexities simply do not fit existing classification schemes, which means that the use of this tool prevents these residual indicators of collaboration "health" from becoming visible.

The classification scheme embedded in Rocky's agile-based toolset were not in sync with nor did they include qualitative aspects of the development work. For example, results from code review, eligibility of technical support, and the applicability of provided documentation was missing from the project review agenda. Instead, reviews relied on tools that calculated the number of hours spent in the sprint, the estimated measures of hours remaining, and the state of PBI 'completeness' (even PBIs with defects). The emphasis on speed—encoded in the tool design and meeting practices—risks de-emphasizing the quality of the project's collaborative structures and actual work. By applying Scrum in instrumental ways (i.e., by setting up criteria only for when a PBI is signed-off as 'done' irrespective of quality criteria) decreases the potential quality of the work being done according to that methodology. Developers would burn hours and claim tasks to be 'done' despite defects or instability in the code. The classification scheme for progress with the backlog's categories "not started", "in progress", or "done" were insufficient, particularly when activities of code review were ambiguously structured. In theory, quality monitoring was supposedly guaranteed by the agile mechanisms built into the day-to-day meeting structure. As such, meeting outcomes came to stand in as a proxy for the real work being (or not being) done on the project.

Rocky had other formal mechanisms for quality assurance (QA). However, due to the fixed, public nature of the system development, the adopted agile development approaches were forced to operate according to some degree of sequential phasing. This may help explain why the QA team operated outside of the main developmental process, leading to the QA tests not being executed in due time. If the QA team had been part of the everyday coordination procedures, the chances of catching problems earlier would have increased. In the daily work of coordinating with the offshore team, important quality considerations around the code deliveries were absent. Quality was not SITA's focus in the way that they organized their global work, nor in the way that they formalized everyday code reviews. The formal organization of work only represented the quantifiable measures of progression and performance, leaving important aspects of work, and by default those performing the work [33, 38, 50], not visibly represented [57] on the project. This proved troublesome and was emphasized by the lack of space for critical voices.

5.3 Discharge of Accountability in Tasks Execution

Accountability is critical for collaborative work [18]. It entails the ways in which cooperative actors display their actions for others to monitor as well as the ways that others act in mutually accountable ways. Being accountable requires actors to acknowledge, through various feedback mechanisms, that they have noticed and accepted the work of others [6]. While concerns about the quality of the code deliveries in Rocky were flagged early on, these concerns were not accepted as valid. An organization's negative political climate can affect how project managers talk about and share their results; in some cases this may mean assuring others that their projects are successful despite showing opposite results [14]. In Rocky, when IT developers raised concerns about quality, they were dismissed and their comments were interpreted as protectionist resistance relating to the preservation of their own careers [51]. In short, the project manager did not demonstrate accountability in accepting, expounding, or acknowledging his colleagues' real concerns.

The potential to demonstrate accountability in global work is also linked to status. Task allocation and code ownership can be a way to sustain high status in distributed work [42]. Those deemed to have oversight of a high-status task that they manage well—in other words, one demonstrates accountability—often expect to receive acceptance or acknowledgement by getting other high-status tasks in the future. Looking more closely at the ways in which tasks were allocated between the onsite and offshore teams in our case, we can see that the distribution was not random. Instead, the task distribution reflects unconscious assumptions about "cheap" and thus underperforming labor. For global work to be successful, participants at all sites need to accommodate their current work practices to support the global distribution [41]. However, in Rocky, distribution of tasks to

the offshore team did not take into account suitability. Instead, our data hint that the onsite participants choose high status tasks and sent less challenging tasks to the offshore team without any acknowledgement that these tasks required special interdisciplinary support and expertise [21]. This skewed distribution of tasks made it difficult for the offshore developers to interpret the implicit knowledge received [36] and, correspondingly, to demonstrate accountability. Further, the lack of officially assigned technical expertise onsite meant that offshore developers were required to find alternative strategies to obtain the required knowledge for them to solve their tasks. As a result, a small number of onsite technical developers engaged in unofficial invisible work [57] to accommodate requests from offshore developers. These onsite developers were neither formally tasked with this responsibility nor were they acknowledged for their everyday effort and support. Yet by informally interacting with offshore developers, onsite developers gained insights into critical technical problems with the code deliveries long before such problems were formally identified.

5.4 No Means Was Found to Align the Requirements of the Business Case, the Work Procedures, and the Coding Tasks

Early estimates of software quality, such as determining where additional code inspections or tests are needed [43], have been found to be beneficial when coordinating software development work. The complexity of the system developed in the Rocky project meant that tasks and sub-tasks evolved over time due to the vast amount of external integrations required. Evaluating the quality was largely an interdisciplinary mixture of diverse quality aspects from domain experts as well as technical experts, not a singular assessment of structured code. Such complex measures could not be captured or supported by the few tools mandated in Rocky because their limited heterogeneity failed to reflect the complexity of diverse quality assessments or considerations of the socio-technical and interdisciplinary knowledge required to organize work across geographical sites [48]. The basic problem was a difficulty in aligning business case requirements, work procedure requirements, and coding tasks. No effort was invested in figuring out how to deal with these issues or how to continuously evaluate project progression. The sparse classification scheme available in the implemented tools and practices, already fundamentally challenged by difficult alignment of agile processes with a rigid yet complex project, was not enough to allow potential problems to become visible in due time.

Assessing the quality of a completed piece of work requires frequent review by diverse experts, especially when moving between project stages. Business analysts' knowledge about business refinements and clarifications needs to be shared initially with the IT developers and then with the technical experts. Measuring quality under such a complex frame must, by definition, take into account both the people who have been formally included in the work as well as those who have been informally involved, that is the 'invisible workers'. Further, it is essential that organizations such as SITA find ways to engage with critical voices within the project. Creating a culture where critical dialogue is encourage would allow for a revisiting of unconscious cultural assumptions [54] as well as basic assumptions about global work in general [45]. This is especially necessary since the assumptions about "cheap labor" and "poor programming skills" create a fragile foundation for global work, where the risk of failure is high. Such an approach requires us to acknowledge concerns displayed by all participants—both onsite and offshore—and to find ways to foster accountability across all of the diverse interlinked groups in a global collaboration.

6 CONCLUSION

This paper details an empirical account of a failed collaboration in a global software development project to understand how it is that problems in global collaborations can go undiscovered until it is too late. Previous research has pointed to the importance of ensuring frequent interaction in closely coupled work in GSD. We extend this research by demonstrating how frequent interaction and closely coupled work have little or even harmful effects when the tools and collaborative practices used in coordinating work and monitoring performance provide only a partial picture of the global collaboration. We found that the adopted agile development tools (sprint backlog and burn-down chart), executed in the form of a mundane tool (a spreadsheet), were incapable of constructively informing collaborators about issues in globally distributed development work. Instead, use of these tools helped to push the complex nuances of the project to the

background. Our work demonstrates the need to consider how professional roles are involved in the frequent interaction and coupled work of global collaborators. In particular, we point to how 'high status' workers can leverage organizational politics to discharge accountability in the tasks they send to the offshore team. The onsite workers' flawed task allocation and the way socio-technical skill and expertise were distributed across sites made it close to impossible for remote team members to succeed in accomplishing their tasks. In addition to the practices and assumptions made about the global collaboration, we return to the mundane tools applied. Here we demonstrate how the tools also failed to take into account socio-technical considerations, coordination challenges, and the invisible work required for effective, distributed work. These tools used categories that reflected the underlying financial incentives of GSD, where global software development work is considered solely a matter of distributing technical tasks across geographical sites and where performance measures such as task execution velocity (speed) is the primary focus.

Finally, project participants were not taken seriously when trying to flag problematic issues in the project. We do not argue that the collaboration failed because of the mundane tools. Instead, we argue that when organizations apply mundane project management tools as the main form of monitoring, there is a risk that the emerging clues indicating a potential derailment are hidden and critical problems are not noticed in due time. Failure was the result of the semantic issues that arose out of distinctive and uncoordinated professional expertise; an unwillingness to confront organizational problems and their consequences (or lack of awareness about those problems); and the fact that – in a global environment – the tools and the collaborative processes, which might indicate the existence of such issues, failed to do so. In our case these contingencies relate to the ways in which work tasks were divided between remote collaborators based (presumably) upon preferences about high/low status work rather than considerations for creating conditions for successfully accomplishing and solving technical problems. As a consequence, our study suggests that everyday coordination practices and tools can render important aspects of software development work invisible in GSD and allow 'high status' workers to articulate biased assumptions. In contrast, we assert that tools should provide a space for critical reflection as a means to assess the appropriateness of one's own local organization of work and tool use.

ACKNOWLEDGEMENTS

Thanks to the anonymous reviewers whose insightful comments and responses helped us improve this work. Moreover, we would like to express a special thanks to Ingrid Erickson and Dave Randall for your invaluable critique and constructive commenting on previous versions. Also thanks Naja Holten-Møller, María Menendez Blanco, Karim Jabbar, Mika Yasuoka, and Robyn Schimmer for taking part in discussing early drafts of this paper. This work could not have been accomplished if it had not been for the anonymous volunteers who generously offered their time and thoughts for this research, so many thanks to all of you. This work was supported by the The Danish Council for Strategic Research and the "Next Generation Technologies and Processes for Global Software Development" research project #10-092313 nexgsd.org.

REFERENCES

[1] Gabriela Avram, Liam Bannon, John Bowers, Anne Sheehan, and Daniel K Sullivan. 2009. Bridging, Patching and Keeping the Work Flowing: Defect Resolution in Distributed Software Development. Computer Supported Cooperative Work (CSCW) 18, 5-6, 477–507. http://doi.org/10.1007/s10606-009-9099-6

[2] D E Bailey, P M Leonardi, and S R Barley. 2012. The lure of the virtual. *Organization Science*. http://doi.org/10.1287/orsc.1110.0703

[3] P L Bannerman. 2008. Risk and risk management in software projects: A reassessment. *Journal of Systems and Software*. http://doi.org/10.1016/j.jss.2008.03.059

[4] Marc Berg. 2001. Implementing information systems in health care organizations: myths and challenges. *International Journal of Medical Informatics* 64, 2-3, 143–156. http://doi.org/10.1016/S1386-5056(01)00200-3

[5] Pernille Bjørn and Nina Boulus. 2011. Dissenting in reflective conversations: Critical components of doing action research. *Action Research* 9, 3, 282–302. http://doi.org/10.1177/1476750310396949

[6] Pernille Bjørn and Ojelanki Ngwenyama. 2009. Virtual team collaboration: building shared meaning, resolving breakdowns and creating translucence. *Information Systems Journal* 19, 3, 227–253. http://doi.org/10.1111/j.1365-2575.2007.00281.x

[7] Pernille Bjørn, Sue Burgoyne, Vicky Crompton, Teri MacDonald, Barbe Pickering, and Sue Munro. 2009. Boundary factors and contextual contingencies: configuring electronic templates for healthcare professionals. *Eur J Inf Syst* 18, 5, 428–441. http://doi.org/10.1057/ejis.2009.34

[8] Pernille Bjørn, Morten Esbensen, Rasmus Eskild Jensen, and Stina Matthiesen. 2014. Does Distance Still Matter? Revisiting the CSCW Fundamentals on Distributed Collaboration. *ACM Trans. Comput.-Hum. Interact.* 21, 5, 27–26. http://doi.org/10.1145/2670534

[9] Jeanette Blomberg and Helena Karasti. 2013. Reflections on 25 Years of Ethnography in CSCW. *Computer Supported Cooperative Work (CSCW)* 22, 4-6, 373–423. http://doi.org/10.1007/s10606-012-9183-1

[10] Alexander Boden, Bernhard Nett, and Volker Wulf. 2007. Coordination Practices in Distributed Software Development of Small Enterprises. IEEE Computer Society, 235–246. http://doi.org/10.1109/icgse.2007.18

[11] Alexander Boden, Bernhard Nett, and Volker Wulf. 2009. Trust and Social Capital: Revisiting an Offshoring Failure Story of a Small German Software Company. In *ECSCW 2009*, Ina Wagner, Hilda Tellioğlu, Ellen Balka, Carla Simone and Luigina Ciolfi (eds.). Springer London, 123–142. http://doi.org/10.1007/978-1-84882-854-4_7

[12] Alexander Boden, Frank Rosswog, Gunnar Stevens, and Volker Wulf. 2014. Articulation spaces: bridging the gap between formal and informal coordination. ACM, 1120–1130. http://doi.org/10.1145/2531602.2531621

[13] Geoffrey C Bowker and Susan Leigh Star. 2000. *Sorting Things Out*. MIT Press.

[14] Narciso Cerpa and June M Verner. 2009. Why did your project fail? *Commun. ACM* 52, 12, 130–5. http://doi.org/10.1145/1610252.1610286

[15] Lars R. Christensen and Pernille Bjørn. 2014. Documentscape: intertextuality, sequentiality, & autonomy at work. http://doi.org/10.1145/2556288.2557305

[16] Kathleen M Eisenhardt. 1989. Building Theories from Case Study Research. *The Academy of Management Review*, 14, 4, 532–550.

[17] Robert M Emerson, Rachel I Fretz, and Linda L Shaw. 2011. *Writing Ethnographic Fieldnotes, Second Edition*. University of Chicago Press.

[18] Thomas Erickson and Wendy A Kellogg. 2000. Social translucence: an approach to designing systems that support social processes. *ACM Trans. Comput.-Hum. Interact.* 7, 1, 59–83. http://doi.org/10.1145/344949.345004

[19] Morten Esbensen and Pernille Bjørn. 2014. Routine and Standardization in Global Software Development. ACM Request Permissions, 12–23. http://doi.org/10.1145/2660398.2660413

[20] Ewusi-Mensah. 2003. *Software Development Failures*. MIT Press.

[21] Samer Faraj and Lee Sproull. 2000. Coordinating Expertise in Software Development Teams. *Manage. Sci.* 46, 12, 1554–1568. http://doi.org/10.1287/mnsc.46.12.1554.12072

[22] DianaE Forsythe. 1999. "It's Just a Matter of Common Sense": Ethnography as Invisible Work. *Computer Supported Cooperative Work (CSCW)* 8, 1-2, 127–145. http://doi.org/10.1023/A:1008692231284

[23] Marco Aurelio Gerosa, David Redmiles, Pernille Bjørn, and Anita Sarma. 2015. Editorial: Thematic series on software engineering from a social network perspective. *Journal of Internet Services and Applications* 6, 1, 109. http://doi.org/10.1186/s13174-015-0038-0

[24] Rebecca E Grinter. 2003. Recomposition: Coordinating a Web of Software Dependencies. *Computer Supported Cooperative Work (CSCW)* 12, 3, 297–327. http://doi.org/10.1023/A:1025012916465

[25] Wen-Ming Han and Sun-Jen Huang. 2007. An empirical analysis of risk components and performance on software projects. *Journal of Systems and Software* 80, 1, 42–50. http://doi.org/10.1016/j.jss.2006.04.030

[26] James D Herbsleb and Audris Mockus. 2003. An Empirical Study of Speed and Communication in Globally Distributed Software Development. *Software Engineering, IEEE Transactions on* 29, 6, 481–494. http://doi.org/10.1109/TSE.2003.1205177

[27] Pamela J Hinds and Diane E Bailey. 2003. Out of Sight, Out of Sync: Understanding Conflict in Distributed Teams. *Organization Science* 14, 6, 615–632. http://doi.org/10.1287/orsc.14.6.615.24872

[28] Pamela J Hinds and Mark Mortensen. 2005. Understanding Conflict in Geographically Distributed Teams: The Moderating Effects of Shared Identity, Shared Context, and Spontaneous Communication. *Organization Science* 16, 3, 290–307. http://doi.org/10.1287/orsc.1050.0122

[29] Pamela Hinds, Daniela Retelny, and Catherine Cramton. 2015. In the Flow, Being Heard, and Having Opportunities. ACM Press, 864–875. http://doi.org/10.1145/2675133.2675199

[30] Naja L Holten Møller and Signe Vikkelsø. 2012. The Clinical Work of Secretaries: Exploring the Intersection of Administrative and Clinical Work in the Diagnosing Process. In *From Research to Practice in the Design of Cooperative Systems: Results and Open Challenges*. Springer London, London, 33–47. http://doi.org/10.1007/978-1-4471-4093-1_3

[31] Waqar Hussain and Tony Clear. 2014. Spreadsheets As Collaborative Technologies in Global Requirements Change Management. IEEE Computer Society, 74–83. http://doi.org/10.1109/ICGSE.2014.25

[32] Lilly C Irani and M Six Silberman. 2013. Turkopticon: interrupting worker invisibility in amazon mechanical turk. ACM, 611–620. http://doi.org/10.1145/2470654.2470742

[33] Lilly Irani, Paul Dourish, and Melissa Mazmanian. 2010. Shopping for Sharpies in Seattle: Mundane Infrastructures of Transnational Design. ACM, 39–48. http://doi.org/10.1145/1841853.1841860

[34] Rasmus Eskild Jensen and Pernille Bjørn. 2012. Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds. In *From Research to Practice in the Design of Cooperative Systems: Results and Open Challenges*, Julie Dugdale, Cédric Masclet, Maria Antonietta Grasso, Jean-François Boujut and Parina Hassanaly (eds.). Springer London, 123–136. http://doi.org/10.1007/978-1-4471-4093-1_9

[35] Rasmus Eskild Jensen and Bonnie Nardi. 2014. The Rhetoric of Culture as an Act of Closure in a Cross-national Software Development Department.

[36] Paul M Leonardi and Diane E Bailey. 2008. Transformational technologies and the creation of new work practices: making implicit knowledge explicit in task-based offshoring. *MIS Q.* 32, 2, 411–436.

[37] P Luff, J Hindmarsh, and C Heath. 2000. Workplace Studies: Recovering Work Practice and Informing System Design. Cambridge University Press.

[38] David Martin, Benjamin V. Hanrahan, Jacki O'Neill, and Neha Gupta. 2014. Being a turker. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (CSCW '14). ACM, New York, NY, USA, 224-235. DOI=http://dx.doi.org/10.1145/2531602.2531663

[39] David Martin, Jacki O'Neill, Dave Randall, and Mark Rouncefield. 2007. How Can I Help You? Call Centres, Classification Work and Coordination. *Computer Supported Cooperative Work (CSCW)* 16, 3, 231–264. http://doi.org/10.1007/s10606-007-9045-4

[40] Stina Matthiesen and Pernille Bjørn. 2015. Why Replacing Legacy Systems Is So Hard in Global Software Development: An Information Infrastructure Perspective. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (CSCW '15). http://dx.doi.org/10.1145/2675133.2675232

[41] Stina Matthiesen, Pernille Bjørn, and Lise Møller Petersen. 2014. "Figure out how to code with the hands of others": recognizing cultural blind spots in global software development. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (CSCW '14). 1107-1119. http://dx.doi.org/10.1145/2531602.2531612

[42] Anca Metiu. 2006. Owning the Code: Status Closure in Distributed Groups. *Organization Science* 17, 4, 418–435. http://doi.org/10.1287/orsc.1060.0195

[43] Nachiappan Nagappan, Brendan Murphy, and Victor Basili. 2008. The influence of organizational structure on software quality: an empirical case study. ACM Press, 521–530. http://doi.org/10.1145/1368088.1368160
[44] Bonnie A Nardi and Yrjö Engeström. 1999. A Web on the Wind: The Structure of Invisible Work. *Computer Supported Cooperative Work* 8, 1-2, 1–8. http://doi.org/10.1023/A:1008694621289
[45] Tsedal B Neeley, Pamela J Hinds, and Catherine D Cramton. 2012. The (Un)Hidden Turmoil of Language in Global Collaboration. *Organizational Dynamics* 41, 3, 236–244. http://doi.org/10.1016/j.orgdyn.2012.03.008
[46] Tsedal B Neeley. 2012. Language Matters: Status Loss and Achieved Status Distinctions in Global Organizations. *Organization Science* 24, 2, 476–497. http://doi.org/10.1287/orsc.1120.0739
[47] Gary M Olson and Judith S Olson. 2000. Distance matters. *Hum.-Comput. Interact.* 15, 2, 139–178. http://doi.org/10.1207/s15327051hci1523_4
[48] D Randall, R Procter, Y Lin, and M Poschen. 2011. Distributed ontology building as practical work. *International Journal of ….* http://doi.org/10.1016/j.ijhcs.2010.12.011
[49] David Randall, Richard Harper, and Mark Rouncefield. 2007. *Fieldwork for Design.* Springer Science & Business Media.
[50] N Raval and P Dourish. 2016. Standing Out from the Crowd: Emotional Labor, Body Labor, and Temporal Labor in Ridesharing. http://doi.org/10.1145/2818048.2820026
[51] Johann Rost. 2004. Political Reasons for Failed Software Projects. *IEEE Software* 21, 6, 104–103. http://doi.org/10.1109/MS.2004.48
[52] Kari Rönkkö, Yvonne Dittrich, and Dave Randall. 2005. When Plans do not Work Out: How Plans are Used in Software Development Projects. *Computer Supported Cooperative Work (CSCW)* 14, 5, 433–468. http://doi.org/10.1007/s10606-005-9004-x
[53] Ken Schwaber. 2004. *Agile Project Management With Scrum.* Microsoft Press, Redmond, WA, USA.
[54] Phoebe Sengers, Kirsten Boehner, Shay David, and Joseph 'Jofish Kaye. 2005. Reflective Design. ACM, 49–58. http://doi.org/10.1145/1094562.1094569
[55] Susan Leigh Star and Geoffrey C Bowker. 2007. Enacting silence: Residual categories as a challenge for ethics, information systems, and communication. *Ethics and Information Technology* 9, 4, 273–280. http://doi.org/10.1007/s10676-007-9141-7
[56] Susan Leigh Star and Anselm Strauss. 1999. Layers of Silence, Arenas of Voice: The Ecology of Visible and Invisible Work. *Computer Supported Cooperative Work (CSCW)* 8, 1-2, 9–30. http://doi.org/10.1023/A:1008651105359
[57] Susan Leigh Star. 1991. *Invisible Work and Silenced Dialogues in Representing Knowledge.* Women, Work and Computerization: Understanding and Overcoming Bias in Work and Education., Amsterdam, North Holland.
[58] Anselm Strauss. 1988. The articulation of project work: An organizational process. *The Sociological Quarterly* 29, 2, 163–178.
[59] Lucy Suchman. 1995. Making work visible. *Communications of the ACM.* http://doi.org/10.1145/223248.223263
[60] Anne-Marie Søderberg, S Krishna, and Pernille Bjørn. 2013. Global Software Development: Commitment, Trust and Cultural Sensitivity in Strategic Partnerships. *Journal of International Management* 19, 4, 347–361.
[61] Katie G Tanaka and Amy Voida. 2016. Legitimacy Work: Invisible Work in Philanthropic Crowdfunding. ACM, 4550–4561. http://doi.org/10.1145/2858036.2858110
[62] Ina Wagner. 1993. Women's voice: The case of nursing information systems. AI & society 7, 4, 295–310. http://doi.org/10.1007/BF01891413
[63] B Whittaker. 1999. What went wrong? Unsuccessful information technology projects. *Information Management & Computer Security.* http://doi.org/10.1108/09685229910255160

Received June 2017; revised August 2017; accepted November 2017.

Paper 4

Matthiesen, S. and Bjørn, P. 2016. Let's Look Outside the Office: Analytical Lens Unpacking Collaborative Relationships in Global Work. *COOP 2016: Proceedings of the 12th International Conference on the Design of Cooperative Systems, 23–27 May 2016, Trento, Italy*. Springer International Publishing. 107–122.[10]

[10] Please note that this paper draw on data from Empirical Study no. 2. However, the company names were changed for further protection of the company's anonymity as well as for the blind review process. In this paper MData is referred to as *DanTech*, and ITech as *InData.*

Chapter 7
Let's Look Outside the Office: Analytical Lens Unpacking Collaborative Relationships in Global Work

Stina Matthiesen and Pernille Bjørn

Abstract Global software development (GSD) outsourcing setups are assumed to allow IT developers to work *anywhere* and *anytime*, removing the contextual contingencies of physical location. However, we challenge this assumed flexibility in our ethnographic study on GSD work as we unpack the nature of the collaborative work through the experiences and the concerns of the collaborators in Denmark and in India. We explore the *difficulties* in global work to understand how the everyday work practices in the global collaboration are enacted locally. The study shows how the dissimilarities in the local conditions for work are distinctly tied to the societal infrastructures outside the office, which also shape the work within the office. Reflecting on our analytical approach, we propose three analytical moves to investigate the nature of local contextual contingencies posed by the local infrastructures and impacting global work conditions. We argue that CSCW research on global work should include analytical considerations for how societal infrastructures at the different sites impact how work is accomplished locally in transnational encounters.

7.1 Introduction

For many companies, the outsourcing of IT services and software development is an emblem of today's globalization [1], which has led to an increasing interest in exploring global software development (GSD) practices in computer-supported cooperative work (CSCW) research [2–5]. Seeking to bridge temporal and spatial distance among IT workers in distributed teams, the facets of GSD research broadly cover aspects such as routines [5]; cross-cultural issues [6–8]; the use and development of software development tools, methods, and processes [9]; how

S. Matthiesen (✉) • P. Bjørn
Department of Computer Science, University of Copenhagen, Njalsgade 128, DK-2300 Copenhagen, Denmark
e-mail: matthiesen@di.ku.dk; pernille.bjorn@di.ku.dk

A. De Angeli et al. (eds.), *COOP 2016: Proceedings of the 12th International Conference on the Design of Cooperative Systems, 23–27 May 2016, Trento, Italy*,
DOI 10.1007/978-3-319-33464-6_7

software bugs and defects are resolved and handled [10]; and how knowledge, coordination, and communication are managed in GSD projects [see e.g. 3, 6, 11]. The dedication to understanding the complexities of GSD practices is generally driven by an interest in creating new technologies to support communication and coordination. However, the preconditions for communication and coordination, for example, to take place in transnational collaborative environments (such as GSD) have received less attention. In fact, we know little about the contextual contingencies—present at the various locations involved—and if or how they impact collaboration. As it is a core interest for CSCW to explore the basic nature of collaborative work, we join others in paying attention to the underlying structures and local situations [12], the power dynamics at play [13], to unfold whether these structures impact the collaborative work situations or the use and design of technologies. In doing so, we argue that the literature has yet to come up with analytical directions that can help capture and unfold issues in global work when collaborations fail—issues that are otherwise neglected, left unnoticed, or reduced to general terms such as "*culture*" [14].

Through a 6-month ethnographic study of GSD organized around the transnational work between Denmark and India, we investigate the differences in global outsourcing practices by comparing the contextual contingencies that shape the work differently between an Indian IT vendor and its customer, a Danish IT company. One particular interview set us on this course of research, namely an interview with a tester currently working out of India, but who had spent 6 months working onsite in Ballerup, Denmark. She said:

> [. . .] its two different cultures and its two different worlds. We have our own set of difficulties here [in India]. [. . .] I feel it is very easy living there [in Denmark], while it is difficult living here. (*11/27/13, Interview, Tester, India*)

We found it intriguing that the tester expressed how working out of Bangalore was more difficult than working in Ballerup, thus we asked ourselves: What makes it easier to do *global work* in Ballerup compared to Bangalore? Understanding this experience of *difficulty*, we began to unpack the socio-economic relationship manifested in the outsourcing setup. By the tester's ability to compare her bodily experience of working at each location, she explained how global work is performed in "*two different worlds*". Exploring the nature of the seemingly diverse circumstances for work in Ballerup and Bangalore, our attention was directed at understanding how the location of *your body matters* when engaging in global work. Our analysis took us on a journey from unpacking ethnographically how central language constructs about global work are performed in practice toward moving outside of the office to include considerations on the infrastructural aspects of the particular cities involved in global work.

On the basis of our ethnographic inquiry and the questions above, we then ask: *How do the local contextual contingencies shape the conditions for collaboration across sites within transnational work*? We find that unpacking the nature of the collaborative work through the bodily experiences and concerns of the software developers requires us to explore the physical contextual contingencies as they

manifest in everyday work practices. Also we find that even though collaborative dissimilarities across sites emerge when observing the work inside the office, these issues may be grounded in societal circumstances and infrastructures outside the office walls, which, in a recursive relationship, also shapes the work that takes place inside the office. Based on our findings, we propose an analytical lens consisting of three moves to unpack collaborative dissimilarities in transnational work. Stipulating explorations both inside and outside the transnational offices, the first analytical move examines the local work inside the office in the light of common language constructs for describing transnational work. We show how language constructs are important, as they create certain assumptions about the work and thus impact the nature of the work across sites. The second analytical move then encourage us to step outside the office and explore how the underlying assumptions about global work are enacted locally and related to the contextual contingencies emerging from local societal infrastructures. Finally, we move back into the office, bringing with us insights from outside to re-consider the nature of work as it happens within offices.

7.2 Related Work

Recent interest in CSCW research explores the role of technology in global work and life—taking into account the challenges of diverse technological cultures, economic disparities, and digital divides. Research has examined the work of Turkers in India and the US, emphasizing the role of power dynamics and local circumstances in global work [15, 16]. Moreover, we have seen investigations on how politics sneaks into the offices of software companies in the West Bank [17], how political activists use social media to organize demonstrations from a Palestinian village [18], or how multi-lifespan information systems seek to support justice after the genocide in Rwanda [19]. All this work is concerned with the ways in which we can understand the relationship between politics and power balances as nuanced measures enacted through several interlinked historic and economic structures related to technology accessibility and infrastructure [17]. Global software outsourcing is clearly a transnational phenomenon, where we witness a large and growing population of Indian global IT developers working in outsourcing setups with IT developers in Europe or the US [20]. Previous research on IT developers working out of India has demonstrated how the identities of the IT developers are framed by located norms and beliefs as they are constructed and situated in particular locations [21]. When exploring the economic, social, or cultural assumptions and the infrastructural differences that arise in global work, we must enact an alternative sensibility on how to explore the reality of culturally located practices [22–24]. Thus, to investigate the differences in work realities shaped by located contingencies in India and in Denmark, we will pay attention to the localized practices of people engaging in the global work.

The collaborative work in GSD can be challenged by the nature of the software development tasks, the information infrastructural constraints in the system under development [25], as well as the underlying structures that lay the groundwork for the day-to-day collaboration through artefacts and technology use [12]. Furthermore, understanding global software outsourcing also requires investigating the infrastructures that locally affect and facilitate the global work at the different locales. In a study of the displaced population in Bangladesh, certain infrastructural experiences were pertinent to understanding how forced mobility impacted the population's access to technology and thus ways of life, pushing them to create workarounds to sustain life [24]. Building upon these insights, we will explore the ways in which infrastructural work shapes the foundation for the collaboration differently in Denmark and in India.

Time and place matter in global work. Previous research has investigated the timely rhythms and patterns in global software development [5, 26] and the collaboration and negotiation of time across time zones [27]. When exploring the collaboration of global software development, we need to take a close look at how time is organized and decided upon—and by whom. Although work time has been argued to be more flexible in the globalized world, time and place are also related to status and power [13]. Poster [28] shows how workers in Indian call centers face a work-time rigidification and standardization, as the work day is shifted from day to night to sync with consumers' daytime in Europe or in countries such as Japan or the US. She refers to the transformed work time as reversal of work time, which is a pertinent part of the work conditions [28]. Although the demand to work at night is different between workers in call centers and our global IT developers, there are important aspects of time and place that deserve further investigation. When understanding transnational collaborative relationships in relation to time, we found Sharma's work on the taxi driver's relationship to time useful for understanding the politics of time and laboring within temporal infrastructures. Sharma introduces the term cab-lag, which "*refers to a condition of labor where people exist in a differential and inequitable temporal relation with another group with whom they are expected to sync up*" ([29], p. 79), which may be an interesting condition in relation to the global IT workers.

7.3 Method and Location

Since 2011, we have conducted several ethnographic studies of global software development in different parts of the world as part of a large research project (Next Generation tool and processes for global software development—NextGSD). While each study is uniquely organized with a different purpose and aim, they have all provided us with insights into the practices of GSD in general. Throughout these years, we have spent time in India as well as in the Philippines studying global IT development. In particular, we studied IT developers in India working in outsourcing setups collaborating with European and American clients. In 2013 we initiated

Table 7.1 Data sources from fieldwork conducted in India (IN) and Denmark (DK)

Field site			
Gathering technique	IN	DK	Online
Observation (no./hours)	17/9.1	29/35.6	26/14.3
Interviews (no./hours)	7/5	10/8.8	4/2.6
Field diary entry (no.)	9	39	–
Time spent in field (hours)	66	137	–

a new ethnographic study of an outsourcing setup between an Indian IT vendor (which we call *InData*) and their customer (called *DanTech*). DanTech is a large IT and software company in Denmark, with more than 3200 employees and several branches around Denmark. For more than 40 years, DanTech has developed IT products for both the Danish public and private sector, and since 2005 they have conducted GSD outsourcing projects with InData to offer customers a reduced time to market and to ensure extra resources and growth. In 2013, more than 200 people from five different global suppliers located in Poland and India collaborated with DanTech, where InData is the largest supplier.

The ethnographic study [30] was organized as a workplace study [31] conducted both in Ballerup, Denmark, and in Bangalore, India. The first author spent intensive periods in the field—following the IT workers in their daily work, including observing online and collocated meetings—with frequent reflective breaks at the university. While in the field, activities such as observation, note taking, and shadowing were combined with daily field diary, photographs, and semi-structured interviews. We also conducted formal interviews face-to-face or in online meetings through the company's communication platform, Lync—ranging from 20 min to 1 h (all audio-recorded and verbatim transcribed). Moreover, we retrieved documentary evidence of various company artefacts and internal email correspondence. Our data sources collected in Denmark and India cover 59 h of observations, 21 interviews, and a field diary (cf. Table 7.1), and in total we spent around 200 h in the field.

7.4 Results

7.4.1 Bodies Inside the Offices

During our ethnographic fieldwork the dissimilarities in the relationship between InData and DanTech became noticeable in various ways, for example, the agency to decide on the course of action, which remains at the client side in Ballerup, as DanTech manages the project. Witnessed through several observations and frequently articulated by the IT developers working out of Bangalore was that they often find themselves in an idle position waiting for a response from Ballerup in order to work.

> Everything is very time consuming here, I feel . . . over there [in DanTech] everything is fast, it's very quick there. Whatever we do, we get the responses, the answers very quick. Here we have to wait [. . . it] is dragging the tasks [. . .] and that is not the only thing, if we are expecting some answers, if we have sent an email, so we wait for further reply. But of course we can take up another task and do but still if it is of importance or if it is blocking [other work]. That's one of the drawbacks. (*11/27/13*, *Interview*, *Tester*, *India*)

The constant waiting and inability to assume control over time create different conditions for global work in Bangalore compared to Ballerup. Even if work was slowed down at InData, this would rarely block the DanTech IT workers from continuing to work on other tasks, but the opposite impact is always present. In addition, we saw how opposed the practices around meetings and schedules were when DanTech tried to control the organization of work with InData:

> Jakob comes to my desk to inform me about the meeting that we were supposed to start now [. . .]. Murali is not there right now, so we will have to wait a little (he tells me with an eloquent smile on his face that refers to our talk earlier about how he had invited Murali to the meeting and he had not yet accepted it). According to Jakob, when the meeting invitation remains unanswered that does not necessarily mean that he [Murali] will not be attending. Instead, Jakob tells me that he often experiences when inviting one of the developers in India to a [online] meeting that they rarely accept the meeting invitation. This does not mean they will not attend. Instead, as Jakob sees it, the Indians do not see the invitations as a negotiation process in the same way as the Danes do: inviting, accepting, rescheduling, etc. in their online calendars. (*07/05/2013, Fieldwork notes, Denmark*)

The situation above with Jakob and Murali was one of many similar situations. Clearly schedules are practiced differently; at DanTech IT workers are expected to be in office when the calendar says so, they are expected to accept or decline meeting invitations, and they are expected to be precisely "on time". We noticed how meticulously the workers at DanTech used their online calendars to register all work activities, leaving colleagues no trouble in detecting whether a person is available or, for example, attending a meeting. The calendar invites are perceived as an opportunity for negotiating time at DanTech, whereas at InData calendar invites come across as commands and not negotiations. You do not negotiate commands, but rather find other workarounds to get out of non-suitable situations:

> At some point during the daily scrum meeting it came up that Ravi is not here today. Martin sounded surprised by this information. Later I hear Martin telling Lisa that it is annoying that on Friday he and Ravi had discussed something that Ravi should be working on, and then today, Monday, they find out that Ravi is not here. [. . .] Ravi was taking a day off. (*07/08/2013*, *Fieldwork notes*, *Denmark*)

When Martin and Ravi talked on Friday about getting something done on Monday Ravi knew he would be away that day; however, instead of accepting or declining calendar invites Ravi's strategy avoided participating in the negotiation of time and rather than address the issue directly he simply said nothing and stayed home as planned. From Martin's perspective, taking a day off is legitimate; however, Ravi's failure to tell Martin he would be away that day and the lack of transparency in work time is problematic. What we see here is not simply a lack of transparency in the work; the situation demonstrates the performativity of time in the global work.

There is obviously an asymmetry in the relationship between InData and DanTech, and Ravi's lack of power in scheduling and coordinating his own work demonstrates that agency for negotiation of time plays an important role in global work. Had he initiated a discussion on re-scheduling the meeting, it might have put him in an awkward situation of either not being able to satisfy the 'customer' or causing him to cancel his day off. By ignoring the matter of participation in the scheduled meeting, Ravi can take his day off.

7.4.2 Bodies Outside the Offices

The challenge of time was also pertinent outside the office. A tester working out of Bangalore explained: "*We have to drive or come from a long distance*, *and we have to be here for nine-and-a-half hours*." The daily commute in Bangalore, combined with the fact that the InData staff work 2 h more every day compared to DanTech, creates different work conditions related to time. Most foreigners would find the Indian road traffic noisy, complicated, frenzied, and slow. Travelling a distance that might be estimated without traffic to take about 20 min might easily last more than 2 h during rush hour. In Bangalore—the Silicon Valley of India—most of the IT companies are located outside the city in industrial parks or campus-like spaces. InData is situated in Electronics City—one of India's largest electronic industrial parks—and due to the complexity of the transport infrastructure, most of the companies in Electronics City offer their employees bus transportation each day for a monthly fee. The bus transportation is organized by picking up employees at certain times and places and driving them to and from work. InData has more than 23 different buses with seats for 40 people in each. Buses run twice in the morning, striving to arrive at the office campus at 9:00 a.m. and 10 a.m. In the evening, buses leave twice, at 6:30 p.m. and again at 7:30 p.m. Because time to commute in Bangalore is highly unpredictable, coming in "on time" or coming home "on time" are variables produced by the local circumstances and less by the individual. Thus, some IT developers choose alternate commuting practices, traveling at less jammed hours:

> [...] Usually Rati does not go by bus, but because of visits from DanTech today, she will come into the office at 9 a.m. Her normal routine is to take a company car—a car provided by DanTech [...] the car will leave at 11 a.m. to reach the office around noon, and leave again at 10 p.m. to reach her home at 11 p.m. That way Rati will then skip spending time in the heavy traffic, and it is also very good for her, as a tester, to work at the same time as her colleagues onsite. (*12/04/2013*, *Fieldwork notes*, *India*)

According to Rati, the staggered working hours are preferred, although not required, by DanTech; however, she really appreciated the benefits of skipping the heaviest traffic. She further explained that getting into the office around noon and leaving work late suited her lifestyle for now while she is unmarried.

Commuting in Ballerup appears quite different. While the IT developers at DanTech commute every day, foreigners will be surprised to see the regularity

and punctuality by which people arrive each morning between 7:30 a.m. and 9:00 a.m by bike, train, bus, or car. The typical work day is 7–8 h, and most IT developers will leave around 3:00–5:00 p.m. Transportation is articulated by several of the IT developers in DanTech as an opportunity to get exercise and not simply as commuting. Thus, many travel by bike and then shower and dress in the company locker room before walking into the office and booting up their computer. Most roads have dedicated biking lanes, which goes well with the widespread trend of exercising to and from work, wearing full cycling clothes, and riding an exclusive racing bike. The public transportation in Denmark is generally reliable and predictable, and it is possible to plan transportation in detail, including calculating the time it takes to transfer between trains or buses as well as the time it takes to walk to the office. Also, the trains and buses run in routine patterns on a regular schedule, including every ten minutes during rush hour, allowing the IT developers to flexibly plan and organize their travel between home and work.

Besides time, place also emerged as an inevitable object of inquiry in our search to understand the differences in global work at the two locations. In Ballerup we experienced how those with families make sure to plan and customize their time spent physically at the office by taking advantage of flex-time agreements, which means that they can distribute their work hours flexibly over a time period (typically across a month or year) and chose to come into the office later or leave earlier as long as work hours are carefully registered in advance in the online calendar. Thus, the location of the workspace is not necessarily physically bounded; instead, work might take place at different places.

All DanTech workers are provided with a laptop that they can carry around and bring to meetings or the like. When the IT worker returns to her/his sit-to-stand desk, s/he simply places the laptop in the installed docking station and continues working using the desktop screen, keyboard, and mouse. The laptop not only facilitates movement around the office workplace, it also enables the IT workers to, for example, work evenings from home if, in the daytime, they have to accommodate family emergencies or the like. Thus, in many ways, the location of the workspace is not only flexible and non-fixed in time, but also in space. In this way the work for the IT workers at DanTech was *not* characterized by clear demarcation between work and home. Instead, the work caused by the global circumstances in working across time zones increased the lack of work/life boundaries for the individual, who might choose to wake up early to answer emails in bed before breakfast and getting kids to school to accommodate the remote colleagues waiting for answers.

To overcome the infrastructural challenges of transportation in Bangalore, some IT workers choose to move out to live near the workplace. By near we mean significantly shorter time spent on daily commute, for example, from 1–2 h by bus or car twice a day, to 20–30 min spent walking or going by car. However, leaving one's family or immediate community behind to be able to move closer to the workplace may come at a price, and often these rapidly created housing facilities are dull and characterless. People will stay there during workdays and on weekends they will head back to the villages where their families live.

We also became aware of a certain way of living that was particularly popular among the female junior IT developers and testers at InData, namely living at women's paying guest hostels. Several of the young and unmarried women at InData live at these women's hostels, located walking distance from the workplace. Here they do not have to attend to the daily chores of cooking and cleaning. Instead, they can fully focus on work, while living in a safe environment—side-by-side with "travelling companions" that one can walk with when moving on foot to and from the office. Of course, this kind of living situation fits certain types of workers—those who have not yet established a family of their own or those who may live separate from their family during workdays because the family lives in a village far away. In this way, a clear demarcation between home and work is drawn up for the global IT developers in Bangalore. The majority of their time is spent "away" from home and family, and when they are away they engage only with other "away" colleagues, even in their "free" time.

A final and recurrent theme that demonstrates a form of asymmetry in the global collaborative work concerned the rotation plan for the IT workers to travel abroad. For the InData workers, to go "onsite" (i.e., travel to Denmark and work at DanTech for a period of time) is considered a great opportunity:

> [...] the people working offshore, they always aspire to go and work with the client onsite, and so it's always good to give people an opportunity, even if it is a short one. [...] to go abroad and work there [onsite at DanTech] we can earn some good money, do some good work, see around, and comeback, so it's a big motivation factor. (*12/3/13*, *Interview*, *IT developer*, *India*)

While it is a motivational factor for the Indian workers—boosting their CV and getting monetary benefits—going abroad to work onsite also challenges the family life. Thus, the length of the stay may be critical depending on the worker's marital status: "[...] *if he is a bachelor*, *then no problem moving for six months or nine months*". According to the Delivery Manager at InData, traveling abroad is problem free when the workers are unmarried, as compared to when having a family. However, even when married, some IT workers make sure they can travel abroad when the opportunity arises, like this InData tester: "*I asked my husband and he said if it is for a short term*, *then its good*, *so even I can get some opportunity to learn and meet new people. So I took this and went for six months*".

During our fieldwork we noticed tensions related to the rotation plan, as some of the workers felt they were treated unequally:

> When waiting for the elevator going to the 7th floor for lunch, Anjeneya says—out of thin air—that the developers who are getting the opportunity to go onsite and those who are not are very unequally divided. He says that he thinks that the rotation should be more frequent instead of keeping only a few people onsite for two years or more [...] That way more people would be knowledgeable about the business. (*12/02/13*, *Fieldwork notes*, *India*)

When we asked a manager at InData about the rotation plan, his answer somewhat aligned with Anjeneya's view on the current rotation plan:

> [...] *rotation should happen*, *but that did not happen—at least Hina traveled there and then came back*, *but after that nothing happened*, *so no one traveled. Vernon*, *Ranjit*, *Paturi*,

> *and Ashish were there for a long time, more than one year. And obviously the team is waiting for some of them . . . will be waiting to get that opportunity* [*to travel onsite*] *I think that some of the people who already left might have stayed here if they had been offered the opportunity to travel abroad*. (12/3/13, Interview, Delivery manager, India)

Around the fall of 2013 half of the offshore team left InData in search of greener pastures. According to the manager, the unequal divide in the rotation plan had an impact on their decision to leave. Interestingly, the prestige of getting the "opportunity" to travel offshore from DanTech to visit and work with colleagues at InData was hard to find in Denmark. Those who worked closely with the offshore workers had difficulties finding the motivation to go, as there were no great incentives for traveling abroad. The DanTech employees would not cash in any monetary benefits, and without being allowed any days of leave or compensation when returning home, they would have to sacrifice the time spent with family to go offshore. The expectations for continuous development are high at both locations; however, both the willingness and the expectation to spend time abroad are unequally distributed due to the dissimilarities, for example, in career incentives and temporal infrastructures, at the different locations.

7.5 Discussion: The Analytical Lens

As we were intrigued by the InData tester's bodily experience of transnational work being more *difficult* depending on which *world* you are located in, we first set out to understand what made it difficult, and then to question how the local embodied experience of dissimilarities and difficulties were shaped by the conditions for collaboration within transnational collaborative work.

First, in understanding the ways in which the collaborative relationships in global work are manifested in the work practices, we *move inside the office*. Here, the first analytical move is to destabilize taken-for-granted assumptions about knowledge and language as they appear in the practices under investigation [32]. The purpose is to unpack supposedly objective and authoritative vocabulary about the work, which creates certain conditions for the people involved [33]. Following this strategy exploring global work, we find that unpacking key vocabulary and rhetoric of global work—such as "anywhere" and "anytime"—demonstrates an interesting starting point for our analysis. The rhetoric that global work transcends geographical boundaries since work can be done anywhere [34] creates certain imaginaries on how work is accomplished.

In our case, we see how the pressured work time is pertinent at both locations; however, the politics of time—the power to control time and aspects of how time is managed—is differently performed. While we experienced the IT developers in Bangalore to often wait passively for tasks to be delivered to them, they would stay in the office for many hours. Although the hours might not be fully productive, they would always be there, ready, as cab-lagged workers "*waiting to be necessary for others' time needs*" ([29], p. 75). This observation points to the type of relationship

that is embedded in the economics of outsourcing practices—the client views the remote IT developers as available resources standing by, ready at all times. The IT developers working out of Bangalore are dependent on the work and knowledge located at the client—leaving them with less flexibility and agency in the negotiation of synchronous meetings [27]. In the structural organization of work, it is difficult for the IT developer at InData to take initiative, to perform independent agency, since knowledge and power lie elsewhere. On the contrary, we saw DanTech workers face challenges in the collaboration due to disparate practices around the meeting schedules, which at times led to an InData worker's non-appearance at an online meeting. When looking into the work within the shared practices and commonalities of the global collaborative work, the variety of circumstances, habits, and routines become apparent as helpful direction for where to look when understanding collaborative work.

At DanTech, agency and knowledge are a core and central part of the work that also impact the ways in which work time is organized. Pressure to perform within time is pertinent in Ballerup. However, rather than waiting for others to take initiative, they experience the time pressure in terms of performing efficiently and knowledgeably to not lose work to people elsewhere. Time is money in Denmark, literally speaking, since the most expensive part of software companies is the employees' salaries. Typically IT developers' contracts state the exact amount of hours per week (typically 37 h), which means that all extra work either means extra salary to those involved or that they accumulate additional vacation time, which by law has to be executed within the current year. This means there are certain organizational structures that support efficient working days and discourage working extended hours. Although work time has been argued to be more flexible in the globalized world, clearly workers in Bangalore face a different practice of rigidification and standardization of time at work [28] than in Denmark. Previous research demonstrated how shifting work hours can result in temporal work patterns being out of sync locally to accommodate the challenges of asynchronous work globally [28, 35]. While our data did not demonstrate a completely out-of-sync situation, it was clear that evenings of week days were dedicated for work, reducing time for family and leisure.

At InData, the time dedicated to work on a daily, weekly, monthly, and yearly basis is quite high: a 10-h work day, 5 days a week, and in many cases a 2-h commute both ways. Fourteen hours a day dedicated to work leaves only 10 h a day for sleep, eating, family, and leisure, meaning you work and sleep most of the time. IT workers in Bangalore are encouraged to stagger their daily working hours to align with overseas colleagues by working from noon to late at night. In practice it meant that the global IT workers in Bangalore, rather than becoming "free" and "flexible" in time, often experienced a hyper-management of time, which controls important aspects of their lives. On the contrary, the norm for many global IT workers at DanTech is to have the flexibility to work from home by simply taking their laptop home and having direct access via VPN to all the information required for their work. In this way, the boundaries between work and private life are getting blurred, and previous research has pointed to the challenge of increased attention on work

at the expense of leisure time [36]. The hybrid organization of work in Ballerup is of a different kind than in Bangalore, as transnational encounters blend into private life (early morning and late evening), and also into the private space. While we do not see migration in terms of moving into, for example, hostels inside the tech-hub in Ballerup or in regular travels to offshore locations, migration becomes an intertwined relationship between the work place and the home on a daily basis.

Bringing your work laptop out of the office is *not* the norm in Bangalore. Anyone who has visited an IT company in India has experienced the hassle of registering laptops by serial numbers in advance, and how upon arrival computers and other technical equipment are scanned by security. Part of the reason for these routines involves proving to the western clients that the Indian companies are professional and can ensure good security: they demonstrate that data is not leaving the premises. However, it also means that global IT workers at InData are not given laptops so they can work from home on a regular basis.

The reasons *why* some IT workers can work from home, while others obviously cannot, is not what is interesting here. Instead, to unpack global work, it is important to realize the existing and various constraints and limitations placed not only by the local work practices within the collaborating organizations but also by the physical contextual contingencies that emerge upon the infrastructures involved locally. In this case, infrastructural aspects of, for example, the Danish data privacy legislation may contribute to our understanding of the dissimilar collaborative relationship. Moreover, we learnt how the rotation of IT workers was imbalanced in relation to incentives and rewards for those involved in the global work, which may be explained through important infrastructural aspects of, for example, the hierarchical structures within the transnational collaboration, or by the difficulties of acquiring immigration visas for India or Denmark.

Moving out of the office, we consider the infrastructural circumstances in which the global work is embedded. Thus, we travel outside of the spaces where we normally study global work (the offices) and include considerations about how life and the *infrastructures outside the office* create certain conditions for work. We explore the infrastructures of the society, which serve as fundamental for the global work. When we report on the bodily experience of commuting in Bangalore on potholed and polluted roads in worn out busses, we are not trying to neglect that long commutes take place all over the world. Indeed, we are not saying that transportation alone is in fact what challenges globally distributed and collaborative work. Instead, we use it as an example to demonstrate how and why studying GSD work in CSCW needs to include the infrastructural issues that provide the foundation for the global work, including Internet access, transportation, childcare, domestic responsibilities, etc. The point it not to evaluate whether these infrastructures are good or bad, but to make visible the emergent dissimilarities in the collaborative relationships embedded within temporal and infrastructural aspects of the geographical locations, which often fall to the background in our analysis.

So in the same way that the cab-lagged taxi drivers have no ability to control time [29], due to certain temporal inequitable relationships and trafficable infrastructures, the inequitable work relationship as well as the infrastructures in Bangalore take

away agency from the InData IT workers to control time and place. Conversely, for the IT workers at DanTech who have some capability to control time due to, for example, the infrastructural circumstances allowing them to work from a laptop, the separation between work and private life is erased. Increased by the many hours of commuting in Bangalore, the move between places for life and places for work becomes separated in both time and space. We saw how the IT developer *migrates* into the tech-work environment in Electronics City in Bangalore, away from family and friends. The migration [28] becomes a fact when the global IT developers leave their homes and families to travel to the workplace and the workplace becomes the dominating activity in their lives. Our case demonstrated, for example, how women choose to leave home and live in women's hostels closer to the workplace to reduce the time spent on transportation. While this *move* reduces time spent commuting to work, it also increases time required for travelling to their home villages on weekends and holidays to visit family. The migration experience between life and work becomes more pertinent.

We found that global work based upon transnational encounters is rather hybrid engagements where at both sites the intensity in attention toward work is increasing based on how time is practiced. Clearly the politics of time take different forms—coarsely outlined as either long (at times in idle) hours or as fragmented hours with a constant pressure for efficiency; all depend upon the infrastructural foundations, some of which make certain aspects of work possible while others do not.

7.6 Conclusion

In this paper we set out to unpack the nature of global work through the concerns and the bodily experiences of the IT workers within the collaborative relationship of a GSD setup. In particular, we were intrigued to understand why the difficulties of doing global work—among others—depended on the embodied experience of being at certain physical locations. Triggered by our wonderment and as a first analytical move, we questioned the common language constructs that assume GSD work to allow the flexibility of working anywhere and anytime. When looking into the work practices enacted within the office walls of the global collaboration, it became clear how the flexibility in GSD was merely a matter of how politics of time and place were performed at the various locations. Thus, as a second analytical move, and in order to discover how politics of time and place mattered in the global collaboration, we found it necessary to travel outside the office to investigate the local contextual contingencies and the infrastructural aspects involved. In our third and final analytical move, we returned to the office, now with insights useful for re-considering the transnational collaborative work as it happens within offices.

We found that the place for work is produced differently for the IT workers located in Bangalore and Ballerup, which means that the language construct of "anywhere" is not descriptive for workers in Bangalore. Instead, we saw that the remote client controls the places for work. Moreover, the politics of time were also

enacted differently at the two sites, where IT developers at both locations struggled in different ways. In Ballerup, time became "all the time", while in Bangalore it became "here and now". The dissimilarities in the conditions for work emerge since coordination across sites was required and enacted on the premises of the time of the IT workers at DanTech. The existing asymmetry in the collaborative relationship was clear when an IT worker at InData refrained from answering calendar invites to avoid conflict in taking a day's leave of absence. Finally, the underlying infrastructures in terms of Internet access, transportation, and domestic work clearly placed different conditions for work in both locations. Support from grandparents and paid domestic workers was essential for the IT workers at InData to be able to participate in the transnational work under the conditions of how work was organized. At DanTech, governmental daycare infrastructure was an important part of the work for the IT workers. It also meant that the workday might be cut into two (morning and evening), leaving the afternoon for children, shopping, and cooking. To make this possible, the facility for a home office was an essential infrastructure.

Within global collaborative relationships in GSD there are difficulties and dissimilarities in the work emerging from infrastructural aspects such as trafficable infrastructures, housing possibilities, domestic obligations, technology availability, and flexibility in work hours. Together these infrastructural aspects appear as a multiplicity of relations, which we need to include when exploring particular cases of transnational work. It is important to stress that our proposed analytical lens is not intended as a fixed model. Rather, we see the three analytical moves as an inspiration for opening up transnational studies on collaborative work, bringing into consideration the infrastructural multiplicities embedded within global work as they emerge both inside and outside the office walls.

Acknowledgments The study is conducted as a part of "Next Generation Technologies and Processes for Global Software Development," #10-092313. Go to www.nexgsd.org for more information. We would like to thank all the anonymous volunteers at InData and DanTech who generously offered their time for this research. Moreover, we would like to thank Marisa Cohn, Irina Shklovski, Nina Boulus-Rødje, Lars Rune Christensen, Nanna Gorm, Naja Holten Møller, and Paul Dourish for their contribution in discussing previous versions of this paper, and finally a great thanks to the anonymous reviewers for insightful comments.

References

1. Walsham G (2008) ICTs and global working in a non-flat world. In: Barrett M, Davidson E, Middleton C, DeGross JI (eds) Information technology in the service economy: challenges and possibilities for the 21st century. Springer, Boston, pp 13–25
2. Boden A, Nett B, Wulf V (2009) Trust and social capital: revisiting an offshoring failure story of a small German software company. In: Wagner I, Tellioğlu H, Balka E et al (eds) ECSCW 2009. Springer, London, pp 123–142

3. Christensen LR, Bjørn P (2014) Documentscape: intertextuality, sequentiality, & autonomy at work. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14). ACM, 2451–2460. DOI = http://dx.doi.org/10.1145/2556288.2557305
4. Herbsleb JD, Mockus A, Finholt TA, Grinter RE (2000) Distance, dependencies, and delay in a global collaboration. ACM, Philadelphia, pp 319–328
5. Esbensen M, Bjørn P (2014) Routine and standardization in global software development. In: GROUP'14. ACM, p 12–23. DOI = http://dx.doi.org/10.1145/2660398.2660413
6. Boden A, Avram G, Bannon L, Wulf V (2009) Knowledge management in distributed software development teams – does culture matter? In: IEEE Computer Society, pp 18–27
7. Krishna S, Sahay S, Walsham G (2004) Managing cross-cultural issues in global software outsourcing. Commun ACM 47:62–66. doi:10.1145/975817.975818
8. Søderberg A-M, Krishna S, Bjørn P (2013) Global software development: commitment, trust and cultural sensitivity in strategic partnerships. J Int Manag 19:347–361
9. Hossain E, Bannerman PL, Jeffery DR (2011) Scrum practices in global software development: a research framework. Springer, Torre Canne, pp 88–102
10. Avram G, Bannon L, Bowers J et al (2009) Bridging, patching and keeping the work flowing: defect resolution in distributed software development. Comput Supported Coop Work 18:477–507. doi:10.1007/s10606-009-9099-6
11. Avram G (2007) Knowledge work practices in global software development. Electron J Knowl Manag 5:347–356
12. Matthiesen S, Bjørn P, Petersen LM (2014) "Figure out how to code with the hands of others": recognizing cultural blind spots in global software development. In: CSCW'14, ACM Press, pp 1107–1119. DOI = http://dx.doi.org/10.1145/2531602.2531612
13. Hinds P, Retelny D, Cramton C (2015) In the flow, being heard, and having opportunities. In: CSCW'15. ACM Press, New York, pp 864–875
14. Jensen RE, Nardi B (2014) The rhetoric of culture as an act of closure in a cross-national software development department. In: ECIS 2014, Tel Aviv, Israel, 9–11 June 2014
15. Martin D, Hanrahan BV, O'Neill J, Gupta N (2014) Being a turker. In: CSCW '14, ACM, New York. p 224–235. DOI = http://dx.doi.org/10.1145/2531602.2531663
16. Irani L (2015) Difference and dependence among digital workers: the case of Amazon Mechanical Turk. South Atl Q. doi:10.1215/00382876-2831665
17. Boulus-Rødje N, Bjørn P, Ghazawneh A (2015) "It's about Business not Politics": software development between Palestinians and Israelis. In: European Conference on Computer Supported Cooperative Work ECSCW
18. Wulf V, Aal K, Kteish IA, et al (2013) Fighting against the wall: social media use by political activists in a Palestinian village. In: CHI '13. ACM, New York, p 1979–1988. DOI = http://dx.doi.org/10.1145/2470654.2466262
19. Yoo D, Lake M, Nilsen T et al (2013) Envisioning across generations: a multi-lifespan information system for international justice in Rwanda. ACM, New York, pp 2527–2536
20. D'Mello M (2006) Gendered selves and identities of information technology professionals in global software organizations in India. Inf Technol Dev 12:131–158. doi:10.1002/itdj.20031
21. D'Mello M, Sahay S (2007) "I am kind of a nomad where I have to go places and places"... Understanding mobility, place and identity in global software work from India. Inf Organ 17:162–192. doi:10.1016/j.infoandorg.2007.04.001
22. Irani L, Vertesi J, Dourish P, et al (2010) Postcolonial computing: a lens on design and development. In: CHI'10, ACM, New York, p 1311–1320
23. Philip K, Irani L, Dourish P (2012) Postcolonial computing: a tactical survey. Sci Technol Hum Values 37:3–29
24. Ahmed SI, Mim NJ, Jackson SJ (2015) Residual mobilities: infrastructural displacement and post-colonial computing in Bangladesh. In: Proceedings of CHI 2015, ACM Press (2015). doi:10.1145/2702123.2702573
25. Matthiesen S, Bjørn P (2015) Why replacing legacy systems is so hard in global software development: an information infrastructure perspective. In: CSCW'15, ACM, New York, New York, USA, pp 876–890

26. Massey AP, Montoya-Weiss MM, Hung Y-T (2003) Because time matters: temporal coordination in global virtual project teams. J Manag Inf Syst 19:129–155
27. Tang JC, Zhao C, Cao X, Inkpen K (2011) Your time zone or mine?: a study of globally time zone-shifted collaboration. ACM, Hangzhou, pp 235–244
28. Poster WR (2007) Saying "good morning" in the night: the reversal of work time in global ICT service work. Res Sociol Work. doi:10.1016/S0277-2833(07)17003-5
29. Sharma S (2014) In the meantime. Duke University Press, Durham
30. Randall D, Harper R, Rouncefield M (2007) Fieldwork for design. Springer, London
31. Luff P, Hindmarsh J, Heath C (2000) Workplace studies. Cambridge University Press, Cambridge
32. Anderson W, Adams V (2008) Pramoedya's chickens: postcolonial studies of technoscience. The Handbook of Science and Technology Studies, Cambridge, MA, pp 181–204
33. Mainsah H, Morrison A (2014) Participatory design through a cultural lens: insights from postcolonial theory. doi:10.1145/2662155.2662195
34. Perry M, O'hara K, Sellen A et al (2001) Dealing with mobility: understanding access anytime, anywhere. ACM Trans Comput-Hum Interact 8:323–347. doi:10.1145/504704.504707
35. Shome R (2006) Thinking through the diaspora: call centers, India, and a new politics of hybridity. Int J Cult Stud 9:105–124. doi:10.1177/1367877906061167
36. Gregg M (2011) Work's intimacy. Polity Press, Cambridge, UK

Appendix

10.1 Designing an Implicit Bias Workshop for an Enterprise

The development of the workshop on implicit bias was an iterative process that included presenting and discussing the workshop content with the GD VP at Enterprise IT and the research group at the university. In essence, I wanted to figure out what implicit bias could offer for attempts to move beyond negative stereotyping and to unpack collaborative issues, which are often reasoned to emerge due to 'cultural differences' in global software development (Jensen and Nardi, 2014). Content was developed based on the empirical data collected in all of my three empirical studies, as well data from related GSD studies, the existing body of knowledge on implicit bias, and the hands-on practices for implicit bias training that has been introduced in domains working with people, analytics, and recruitment. The workshop target group was initially the Danish employees at Enterprise IT, as the workshops were planned to take place at two different office locations in Denmark. However, we posted the call for participants in both English and Danish on *Yammer* (the enterprise SoMe platform), which includes all locations in Enterprise IT. In that way, seats were open to participants from other countries; however, given the time required for traveling, the two-hour workshops were less attractive to employees located in, for instance, Sweden or Norway.

One of the motivations for launching the workshop was to help the GD department in addressing the very common issues that exist around GD and software offshore outsourcing. In particular, the workshop was designed to help address the challenges involved with changing work practices from working in a co-located way to a geographically distributed way and the uncertainty about future job security for the employees working at locations where the involvement of GD was being implemented. In this way, offering a workshop was a way to signal that the GD department was open to discussing with and learning from the people involved in GD projects on a daily basis. Moreover, I had identified an increasing need—on the part of the organization—to understand the role and responsibility of the GD department and thus, the GD department decided to go on a tour to visit the various locations to improve communication and to become more visible as a department

within Enterprise IT. Rather than only operating from the headquarters, the GD department needed to become more visible and sensitive to the many different voices related to global work that exist outside of the headquarters. To structure and align these agendas, I formulated one core question, which became the guiding principle in how to address the issues surrounding GD in Enterprise IT: "*How can we create better conditions for global collaboration in Enterprise IT?*" What is important to state with this question is that we did not want to focus on rehearsing the stereotypical concerns regarding global work; the intention, instead, was to invite people to join the discussion around expanding the view on GD work and creating appropriate structures for supporting global software development. This new perspective was based on the concept of implicit bias, and the aim was to improve the ways in which people in Enterprise IT **experience** and **understand** '**cultural problems**' in global collaboration, and to improve the **conditions for global collaboration** in relation to the ways in which cultural differences are dealt with.

The selection of our target group should not be interpreted as implying that the people employed at the Danish locations are more inclined to be implicitly biased and to create cultural stereotypes than the employees in Norway or Sweden, or the GD consultants working from Poland, India, or Ukraine. Rather, this selection was based upon insights from my prior studies, which demonstrated how companies that start to engage in global outsourcing often lack attention toward their existing staff (Matthiesen et al., 2014; Matthiesen and Bjørn, 2017). Thus, given that the Danes were witnessing changes in the collaborative practices—moving from supporting local software development to global software development—the entry point for the Danish employees was different than that of the Polish consultants, for example, who were recruited into the company to engage in globally distributed collaboration from the very beginning. Indeed, the GD department could also have launched these workshops in Norway or Sweden; however, as I was operating from Denmark it was a natural choice to create the trial within Danish office locations.

When organizing the workshops, we planned on doing a workshop in the morning at one location, and another workshop at a second location in the afternoon, each with up to 10 participants enrolled. The participants signed up voluntarily for the

workshops and the only criterion was that the participants had experience with globally distributed collaboration when working for Enterprise IT or in their previous jobs. Unfortunately, only two participants (both managers) wanted to join the morning workshops so, upon the managers' request, we decided to cancel the morning workshops and instead pay one of the managers an informal visit to discuss GD in Enterprise IT. For the afternoon workshops, we received a good response and enrolment from middle management; however, there were fewer participants from what we initially considered as our target audience, namely the employees working as IT architects, IT consultants, software developers, and testers. This misalignment between what we (in the GD department) expected to be the target audience and the goals that we had with the workshop may point to a (implicit) bias that I as researcher within the domain and organization have become accustomed to applying when encountering challenges with globally distributed work. Seven participants joined the workshops and had various positions in Enterprise IT: managers (senior, project, resource, people) (4), a managing director (1), a functional architect (1), and a business developer (1). The workshops lasted two hours each, and entire sessions were documented through field notes, and video and voice recordings. We ran two afternoon workshops with the same group of participants (with a few cancellations due to the winter holiday in Denmark). The first workshop took place one month after the second workshop.

Photo 2: Implicit Bias Workshop, Denmark, Study 3

Appendix

The workshop content ended up entailing introductions to the concepts of implicit bias and confirmation bias. Moreover, I introduced a case on GSD (Matthiesen and Bjørn, 2017) to demonstrate the commonalities in the challenges they meet with regard to globally distributed work in the software industry. The participants were asked to do two group exercises at the first workshop: a personas exercise and an IAT test conducted in the plenary. Moreover, the participants were asked to fill out an online questionnaire prior to both the first and second workshops, and for the second workshop the participants were further asked to bring examples that took into consideration situations where implicit bias comes into play in their everyday work practices, team collaboration, or in their use of collaborative tools. The participants were encouraged to bring back photos of a collaborative situation or of a collaborative tool from a work situation they had reflected upon in relation to implicit bias. The feedback we received during the first workshop and from the questionnaires showed that the participants felt that there had been too little time to engage in general discussions on GD in Enterprise IT. In particular, the participants expressed how they found the organizational structures and systems insufficient for supporting GD in Enterprise IT. The participants expressed how it was hard to figure out how to realize the official corporate vision and GD strategy in practice, and thus for the second workshop we set aside much more time for plenary discussions.

Before the second workshop, participants were asked to fill out another questionnaire to evaluate the first workshop. Moreover, we asked the participants to reflect upon whether they saw any value in boosting their attention toward implicit bias. Thus, the starting point for the second workshop was to have the participants discussing and reflecting upon the observations they had made in relation to implicit bias since the first workshop in their own or their colleagues' work and practices. They were encouraged to consider the efforts and engagements they contributed themselves, the contributions from Enterprise IT and the GD group, and, finally, the tools they used in their everyday work with GD.