

Translocality in Global Software Development

The Dark Side of Global Agile

Bjørn, Pernille; Søderberg, Anne-Marie; Krishna, S.

Document Version

Accepted author manuscript

Published in:

Human-Computer Interaction

DOI:

[10.1080/07370024.2017.1398092](https://doi.org/10.1080/07370024.2017.1398092)

Publication date:

2019

License

Unspecified

Citation for published version (APA):

Bjørn, P., Søderberg, A.-M., & Krishna, S. (2019). Translocality in Global Software Development: The Dark Side of Global Agile. *Human-Computer Interaction*, 34(2), 174-203. <https://doi.org/10.1080/07370024.2017.1398092>

[Link to publication in CBS Research Portal](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us (research.lib@cbs.dk) providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 18. Jun. 2025



Translocality in Global Software Development: The Dark Side of Global Agile

Pernille Bjørn, Anne-Marie Søderberg, and S. Krishna

Journal article (Accepted version)

Cite: Translocality in Global Software Development : The Dark Side of Global Agile. / Bjørn, Pernille; Søderberg, Anne-Marie; Krishna, S. In: *Human-Computer Interaction*, 13.12.2017, p. 1-30.

This is an Accepted Manuscript of an article published by Taylor & Francis in *Human-Computer Interaction* on 13 Dec 2017, available online:

<http://www.tandfonline.com/10.1080/07370024.2017.1398092>

Uploaded to [Research@CBS](#): November 2018

Translocality in Global Software Development:

The Dark Side of Global Agile

Running Head: Dark Side of Global Agile

Pernille Bjørn, University of Copenhagen, Denmark. Email: pernille.bjorn@di.ku.dk

Anne-Marie Søderberg, Copenhagen Business School, Denmark. Email: aso.msc@cbs.dk

S. Krishna, Indian Institute of Management Bangalore, India. Email: skrishna@iimb.ernet.in

ABSTRACT

What happens when agile methods are introduced in global outsourcing set-ups? Agile methods are designed to empower IT developers in decision-making through self-managing collocated teams. We studied how agile methods were introduced into global outsourcing from the Indian IT vendor's perspective. We explored how agile processes in global outsourcing impacts work conditions of the Indian IT developers, and were surprised to find that agile methodologies, even after three years of implementation, created a stressful and inflexible work environment negatively impacting their personal lives. Many of the negative aspects of work, which agile methodologies were developed to reduce, were evident in the global agile outsourcing set-up.

CONTENTS

1. INTRODUCTION
2. GLOBALLY DISTRIBUTED WORK & TRANSLOCALITY
 - 2.1. Globally Distributed Work
 - 2.2. Translocality
3. EMPIRICAL STUDY
 - 3.1. The Indian Global IT Vendor
 - 3.2. Empirical Material
 - 3.3. Analytical Approach
4. RESULTS
 - 4.1. From Waterfall to Global Agile
 - 4.2. Continuous & Out-of-Sync Work
 - 4.3. Agile as Global Micromanagement
 - 4.4. Stressful Work Environment
5. DISCUSSION
 - 5.1. Global Outsourcing in Transnational Spaces
 - 5.2. Translocality in Globally Distributed Work
6. CONCLUSION

1. INTRODUCTION

Distributed collaboration has been core for HCI research since the late nineties (Olson and Olson 2000) and is still a central research domain (Bjørn, Esbensen et al. 2014, Olson and Olson 2014, Esbensen, Tell et al. 2015). One of the problems to tackle in this type of research is how to unpack the complexities arising from transnational encounters, which are part of the embedded nature of most distributed collaboration in global organizations today. Aspects such as discontinuities in language use, work practices, and technologies – just to refer to a few – exist and impact the collaboration (Watson-Manheim, Chudoba et al. 2002, Hinds, Neeley et al. 2014). When we meet at conferences, those who study global collaboration all agree that collaboration between participants located in the ‘global South’ and in the ‘global North’ creates specific work conditions, which are important to understand. Still we lack the vocabulary for how to talk about it, and bring it into the center of our analysis (Bjørn 2016). Often when we explore the economic and political challenges arisen in the transnational encounters, we lack the wording and analytical lenses to guide us. Different attempts have been made to capture this complexity in terms of power asymmetry in distributed work (Hinds, Retelny et al. 2015) or bring in the perceived status differences and the societal infrastructures, which exist locally, into the analytical frame (Levina and Vaast 2008). However, previous literature has not focused on the impact new software development methodologies have on the work conditions of the employees working ‘offshore’, in offices at great distance to clients’ headquarters. In this paper, we study a particular context of distributed work – namely outsourcing of software development. Software outsourcing is characterized by client companies outsourcing software development tasks to IT vendor companies, who have employees working from other locations than the client company, taking advantage of the differences in salaries between low-income and high-income countries with the aim of reducing costs while having access to highly skilled software engineers. Thus, when we study the impact of introducing agile in global outsourcing, it is important to notice that both the client-vendor relationship and the cost-reduction agenda play important roles. We cannot separate out the three entities: The agile methodology, the global work, and the client-vendor relation. Instead we must study the dynamics and relations between the three as they emerge in our empirical material and findings.

The foundation for this paper is an empirical study of global outsourcing conducted within a global IT vendor organization of Indian origin, in this paper anonymized as Global Software Services (GSS). The study took place in the period between December 2011 and February 2014, where we conducted interviews with employees in three large IT projects. Interviews were primarily conducted in India, but also at vendor sites in United Kingdom, Hungary, and Denmark. GSS has been in the software business for more than 40 years and has a strong reputation in global delivery based on experiences in organizing large complex IT projects for international clients. The empirical work provided us many unique insights into a diverse set of aspects important for understanding global software development practices (Søderberg, Krishna et al. 2013, Søderberg 2015, Søderberg and Romani 2017). Over the years of our study, GSS went from global delivery applying waterfall methodology to global delivery applying agile methodology, which made us interested in exploring the impact, which the introduction of agile methods has on global software development in a client-vendor relation, and

more importantly the impact on the work conditions for the IT developers working out of India, whom we interviewed three times during our study.

In this paper, we introduce translocality as an analytical approach for examining globally distributed work. In particular, translocality makes us explore how the use of tools and methods shapes the performance of transparency and invisibility in certain important ways. Addressing translocality in empirical cases includes scrutinizing the potential asymmetric relations embedded in the collaboration through manifestations in tools, methods, and practices. We argue, that we, as HCI researchers, must be willing to reflect upon how the introduction of certain tools and technologies might carry an agenda counter to designers' intention. Our findings demonstrate that introducing agile methodologies, including agile tools and technologies, in global software outsourcing increases transparency and coordination across distributed actors. Furthermore, we argue that this transparency also introduces a risk of disempowering IT developers and testers in the vendor company's offshore sites by reducing their decision agency, thus negatively impacting their work conditions. Understanding this risk presents a key challenge for HCI scholars engaged in designing technologies, methods and tools to support globally distributed work.

The contributions of this paper are multiple. First, we present unique empirical material on global software development as it is experienced by Indian vendor employees in the transition from global delivery structured by the waterfall model towards the implementation of a global agile methodology. Second, we introduce translocality as an analytical approach to the study of globally distributed work. Finally, we demonstrate how the introduction of agile processes in global software outsourcing projects negatively impact the work conditions of the Indian IT developers, by limiting their space to maneuver and taking away their freedom to maintain a work/life balance. These findings point to how global agile in outsourcing relationships with huge time zone differences risks producing work conditions for highly educated IT developers in the global South that are significantly different from those of the IT developers working in the global North, thus opposing the basic ideas of empowerment and self-management initially phrased in the agile manifesto.

The paper is structured as follows. First, we introduce current literature on globally distributed work in HCI, and then present the theoretical foundation for our concept of translocality. Second, we introduce the empirical case, our method, and data sources. Third, we present the results of our analysis providing examples and quotations demonstrating the transformation from waterfall to global agile in the vendor organization. Finally, we discuss our empirical findings and the larger impact of translocality as an analytical approach in globally distributed work.

2. DISTRIBUTED WORK & TRANSLOCALITY

2.1 Globally Distributed Work

Working remotely in distributed collaborative settings has been and continues to be a key interest of research within HCI since the late 1990s. Seminal research on distributed

work, in the early days of technologies being able to supported collaboration across distance, made it clear that distance matters, and only in situations with a high degree of common ground, low coupling of work, with high technology readiness and collaboration readiness, were there a chance for successful collaboration across geographical distance (Olson and Olson 2000). Later, the distance framework has been expanded to include the organization and management of the work (Olson and Olson 2014), and there is evidence that closely-coupled work tasks actually is possible and in some cases even necessary to make collaboration across distance successful (Bjørn, Esbensen et al. 2014). Frequent communication and interaction is important for collaborative work (Hinds and Bailey 2003, Hinds and Mortensen 2005), and closely-coupled work tasks force collaborators to engage in frequent interaction, since individuals will not be able to do their individual tasks without interacting with others, despite the lack of geographical proximity. Traditionally, research on distributed work was interested in understanding the challenges of working remotely compared to collocated work, and use the insights to design new technologies (Esbensen, Tell et al. 2015). Nevertheless, today, developing large software systems in teams of geographically dispersed nature, where participants collaborate across time zones, cultures, languages, and national borders makes it important to expand the analytical perspective to not only focus on the design of new systems, but also understand how the use of diverse technologies and methods shapes the distributed collaboration in particular ways. The main challenges for developing software in a globally distributed setting is not only of a technological nature, it includes a multi-facetted network of challenges we need to address. Today working as an IT developer means being able to collaborate in globally distributed work, and this makes global software development an obvious domain for investigating distributed work.

Global software development (GSD) has become the norm (Herbsleb 2007, Matthiesen and Bjørn 2015, Sørderberg and Romani 2017), and refers to situations, where teams of IT developers are collaborating and spanning boundaries across locations, time zones, organizations and cultural backgrounds. Research in GSD practices within industrial and organizational settings have explored many areas of the distributed collaboration. We can group the main tracks in GSD research into three areas: Coordination, temporality, and communication. Coordination strategies, mechanisms, and technologies is an important area of research within GSD, where focus is on how IT developers find ways to coordinate, segregate, and aggregate various sub-tasks within the software development projects, as well as how the overall project organization supports the coordination of activities (Boden, Nett et al. 2007) – for example how software developers handle dependencies through recomposition (Grinter, Herbsleb et al. 1999), how developers choose to display their work to others supporting awareness (Souza and Redmiles 2007), or how the documentscape of software projects serve as a mechanism for handling the intertextuality, and the autonomy of the dependencies by producing a frame of reference across the multiple files involved in the project, e.g. code files, requirement files, meeting minutes (Christensen and Bjørn 2014). Boundary spanning has been found to facilitate coordination in GSD teams, for example when enacted by vendor managers with bilingual competences and multicultural experiences who frequently engage in dialogue and negotiations with client representatives (Sørderberg 2015, Sørderberg and Romani 2017).

Another important area of research in GSD is temporality that focuses on the structures of GSD practices. Examples include studies of delay in software task solving, and how being geographically dispersed decreases speed in finishing software tasks (Herbsleb and Mockus 2003), as well as how the sequential structure in work can help IT developers to develop routine in their work supporting closely coupled work across distance (Esbensen and Bjørn 2014). The final area of interest is communication, and here previous work points to the importance of a shared meaning context (Bjørn and Ngwenyama 2009) and the challenges of developing shared language about the IT systems, which were developed for local societal situations by remote IT developers (Jensen and Bjørn 2012, Matthiesen and Bjørn 2015). Insights from this work point to that developing a shared language through relation work (Bjørn and Christensen 2011) and having access to important infrastructures for re-negotiation of protocols for work is critically important (Bjørn 2003) both for effective communication in GSD set-ups, as well as for negotiations (Bjørn and Hertzum 2006). Furthermore for developing trustworthiness among team members across geographical sites (Tøth 2015) and commitment in development projects including both client and vendor employees (Søderberg, Krishna et al. 2013).

Coordination, boundary-spanning, temporality, and communication are all important aspects, when we investigate distributed work in GSD settings. In this paper, we are particularly interested in the organization and management of the distributed work (Olson and Olson 2014), since we want to understand the impact, which new methodologies, such as agile development, have on the work conditions for the IT developers working in global outsourcing set-ups. For this specific purpose, we need a different analytical perspective to add to the literature on GSD and distributed work. We propose translocality as an analytical lens to unpack our empirical material.

2.2 Translocality

Translocality as analytical lens is based within the research on transnational approaches for HCI. Transnational HCI is a research agenda, which seeks to emphasize certain aspects of the globalized world and the contemporary conditions by which ICT enables and constrains people's practices in ways fundamental to design and use of technologies (Shklovski, Vertesi et al. 2014). Transnational HCI research provides a vocabulary and an agenda, which place studies of 'the global' in new ways, by rephrasing certain key concepts, as well as replacing commonly held perspectives and challenging basic assumptions about the impact of ICT in a globalized world. Transnational HCI shares many common interests with critical postcolonial research (Shome 2006, Ravishankar, Pan et al. 2013) and has many similarities with postcolonial computing (Philip, Irani et al. 2012). However, what makes transnational HCI research specific is that it insists on investigating the globalized world with a focus on a new type of work arising from the increasing use of technology worldwide. In this paper, our interest is to study outsourcing of software development through the lens of transnational HCI.

Investigating translocality in global outsourcing requires us to rethink and rephrase the concept of culture used in HCI research. Because of the pervasive criticism of the essentialist perspective on culture within other research fields such as anthropology and

cross-cultural management (Søderberg and Holden 2002, Romani, Sackmann et al. 2011), there is a growing movement within HCI, which has started to question whether understanding culture only in terms of nationality is valid, especially since such a perspective tend to leave out the multitude of cultures involved when people collaborate internationally (Boden, Avram et al. 2009). This has lead researchers to explore new ways in which we can investigate culture as part of collaborative practice, e.g. as a shared meaning context across social worlds (Bjørn and Ngwenyama 2009, Jensen and Bjørn 2012), as rhetoric strategies (Gertsen and Søderberg 2012, Jensen and Nardi 2014), status differences and power asymmetry (Metiu 2006, Levina and Vaast 2008, Hinds, Retelny et al. 2015), or as cultural blind spots (Matthiesen, Bjørn et al. 2014). In this paper, we suggest translocality as an analytical strategy, which can help us unpacking complexities in globally distributed work in HCI.

Translocal practices rephrase the concept of culture by insisting on thinking about cultural practices as *verbs* and *doing* - rather than as a noun. People engage in *cultural* practices, which then become constitutive of what makes the culture. In this perspective, culture is detached from national constraints, and becomes a malleable entity, which is formed in practice through negotiations and thus can only be detected in practice through its various manifestations within artefacts or vocabulary. Studying translocality in globally distributed work, thus, includes careful examination of how the cultural practices within the global work arrangement emerge in the complex situations of collaboration, and which embedded logics these situations reflect. This introduces a different analytical perspective on the ways in which HCI researchers can address cultural issues in global collaboration.

Understanding translocality as a practice within globally distributed teams entails a new conceptual vocabulary, which can assist us when investigating the manifestations of cultural practices as they emerge in e.g. the use of methods and tools. It provides us with a new analytical frame, which can help us re-think the ways in which we approach our investigations of global work. The first shift in our analytical understanding is the shift that replaces the dichotomy of global/local with the concept of translocal (Shklovski, Vertesi et al. 2014). The issue with the global/local dichotomy is that it assumes that, we can distinguish between what is local and what is global, while in reality cultural practice will always be experienced from a certain perspective, thus the global is experienced and enacted through the local circumstances. Studying globally distributed work as translocal practices is about studying the various relationships and associations, which make visible the translocal practices experienced by people involved in the globally distributed work. Therefore, if we want to find the ways in which translocality becomes manifested in the collaboration practices, we must remain in the local perspective, and examine the global through the locally manifested experiences. Our strategy in studying translocality in distributed work is thus to follow associations and connections from the immediate local collaborative situation and place the experience in the larger context of the work conditions. It is about identifying translocality as it emerges in the globally distributed work practices, which are immediately both remote, yet central to the experiences of the people involved. In this way, translocality as an analytical lens follows in the foot-steps of multi-sited ethnography (Marcus 1998), and new contemporary ways to explore computer supported cooperative work (Blomberg and Karasti 2013, Williams, Lindtner et

al. 2014, Bjørn and Boulus-Rødje 2015). Following multi-sited methodologies, translocality in globally distributed work is concerned with the multiple and heterogeneous sites of the collaborative work practices (Bjørn and Østerlund 2014) as well as “people’s situated tactics for collaborating and making sense of shifting global relations, across borders and beyond preexisting social frames” (Williams, Lindtner et al. 2014, p. 80). The analytical strategies we use to investigate how people handle the dynamic and shifting relations through their collaborative practices guided us to study the ways in which participants perform translocality, while enacting global connectivity.

3. EMPIRICAL STUDY

As part of a large interdisciplinary research project Next Generation Technologies and Processes for Global Software Development (nexgsd.org), the authors of this paper conducted a study within a large international IT vendor of Indian origin, in this paper anonymized as GSS. The study was organized in such a way that we followed three projects, which each presented long-term relationships between the IT Vendor and important foreign clients. The three projects were different in structure and contract, but they all were examples of large contract. The three clients were a US bank, and a European Bank, and a governmental insurance project.

The study took place between December 2011 and February 2014. Over the years of the study, each of the projects evolved, changed, and transformed. While it is outside the scope of this paper to account for all three projects, we will in this paper zoom in on one important observation, which sparked our interest concerning the perceived changes related to work technology during the previous years. While the introduction of agile was only in the process of being introduced in the first two rounds of interviews, it was fully implemented when we conducted our third and final round of interviews in the vendor’s site in Bangalore. What surprised us was that the IT developers experienced the implementation of agile as more stressful than working with waterfall methodologies. It did not fit with the rhetoric of agile methodology as self-organizing and providing autonomy and decision power to the IT developers (Ågerfalk 2006). Thus, we became interested in understanding how the implementation of global agile methodology impacted the work conditions of the developers in the Indian vendor company. In this paper, we seek to answer the research question: How does the introduction of agile processes in global outsourcing impact the work conditions of the IT developers working out of Bangalore?

3.1. The Empirical Case

The growth of the Indian IT industry since the 1980s starting as fixing the Y2K problem (Subramanian 2006) have increasingly brought outsourcing contracts over the last two decades (Parthasarathy 2004), and even though there have been speculations that the current software industry in India is maturing (Bhatnagar and Madon 1997) due to increased education and international experiences, it is still rare to find long-term projects based upon strategic partnerships (Søderberg, Krishna et al. 2013). The GSS company has existed for more than 40 years and provides IT services, consulting, and business partner solutions to clients all over the world. The company employs more than 300.000

IT consultants and has a presence in more than 45 countries worldwide. It supports multiple service portfolios including enterprise solutions, business process services, and IT infrastructure services. GSS's clients are diverse and include banking and financial services, insurance, healthcare and retail.

The project, which we focus on in this paper, comprised a large portfolio of smaller projects, which were all directed at the same global banking client of US origin. GSS has approximately 4000 employees working on these projects with this client, where the majority of the employees are located in India at different locations, but there are also vendor employees located closer to the client's headquarters in the US. Examples of project constellations are a team of 100 participants, where 40 are in Hyderabad, 50 in Bangalore, 5 in Kochi, and 15 in Phoenix, Arizona, USA. A project might be a smaller sub-project to the financial system such as Mobile Apps (iOS or Android) supporting digital offers based upon location-based technologies (the customer is physically near an ECCO shoe store and the banking app sends digital offers on shoes to the bank's customer), or it might be larger initiatives such as IT systems, processes, and infrastructures supporting new financial services to the US bank's business clients. GSS has worked with this US American client for more than 30 years and is one of its major IT providers. It is important to mention that IT systems are crucial to banking business, since everything is organized through the use of the IT systems. Thus, all IT banking products are highly linked directly to the business of banking. E.g. a new banking product concerning mobile applications would mean that there is a need to develop the IT application which can ensure this banking product can be implemented in the banking practices.

3.2. Empirical Material

The data sources for this paper comprise the interview material collected in the period between December 2011 and February 2014. The majority of the interviews were conducted in Bangalore during three periods. The first period was December 2011-January 2012, the second period in February 2013 and the third in February 2014. Three researchers (two Danes and one Indian) were present during all the interviews. Besides the interviews in Bangalore, we also visited various sites in Europe (United Kingdom, Hungary and Denmark), and interviews were also conducted at these sites. In this paper, we only draw on the interviews we did in Bangalore.

In total 65 interviews with a total of 42 people were conducted over the years of the study. Eight people were interviewed three times, nine people were interviewed twice, and 25 people were interviewed once. We interviewed top managers and account holders, as well as team leads and junior IT developers. All interviews were transcribed in all details. Also, we collected documents and materials concerning the IT vendor both from the company's own perspective as well as public news and other external material on the projects. Further, we engaged over time with the company through workshops, meetings, as well as other events related to research knowledge communication organized as part of our project. We sent drafts and finished research publications to the people we interviewed as well as to contact persons in the vendor organization to get feedback and to validate our interpretation of various empirical observations.

3.3. Analytical approach

For analyzing the empirical material, we decided to focus on one of the three projects, namely the US Bank project. Analyzing these data formed a process where we first read through the transcripts, and then with the knowledge we already have about the project and the company, we began identifying aspects pointing to the experience of working in outsourcing set-ups in India. These aspects might be articulated directly by the participants or emerge implicitly when participants described different events and activities which happened in the project at different times. When investigating distributed work in HCI, there is a tendency to think about the different challenges which emerge as something related to either the ‘global’ or the ‘local’, and thus the analysis becomes organized as an investigation to identify reasons to why complexities might arise from one of these two ‘perspectives’. However, in the translocal perspective, the constant insistence on the localized perspective stipulate that we as researchers in our analysis identify how our informants expressed their experiences of constantly co-existing in different places (Bangalore and Phoenix, for example) at the same time, and how they dynamically were transforming their work to accommodate concrete situations. This makes us notice particular aspects in the interview accounts, which arise in the analytical process and shape our conceptualization of the material.

By following the association from the translocal practices as experienced by the participants, new links and connections across aspects emerged during the analysis – and finally major themes began to take shape. While our concern was for the translocal conditions for the IT developers working out of Bangalore, it is important to notice that the emergent themes were closely connected to our initial curiosity emerging during the last round of interviews in Bangalore 2014, namely how agile software development created unexpected conditions for the global collaboration. In this way, our research question concerning the impact of agile processes on the work conditions of the IT developers was an important thread during our data analysis. As a result of our analysis important themes emerged that help to explore the impact of agile methodology upon the global work of the IT developers. Below we present results of our analysis.

4. FINDINGS

Exploring the impact of agile upon the global work from a translocal perspective, we first need to examine the overall organizational structure by which the agile work processes are implemented. The US Bank is a large client and is highly dependent upon IT systems to run the ongoing business. While the US Bank has several IT vendors as well as in-house IT development – the particular organizational set-up we focus on is the client-vendor relationship between the US Bank and GSS. GSS has an ongoing contract with the US Bank, which comprises a larger program with several sub-projects of diverse size and scope. Each initiative to a new sub-project can come from either the bank or GSS, however it is always the US Bank that takes the decision on whether to move ahead. The collaboration between the two organizations over time has developed with increasing engagement, which means GSS continuously receives a large number of smaller projects to execute.

While the relationship is strong and has existed over many years, the US Bank does not want to be dependent upon GSS, which means that it strategically uses a multiple vendor setup for technology development. This means that for each IT project, the US bank will choose among several competing IT vendors, based upon their performance, and decide who gets which pieces of the contract and thus how the vendors must work together on a common IT project. No vendor has access to the complete project and the client keeps the control. In this way, the multiple-vendor set-up forces competing vendors to collaborate. As a Business Unit Head explained it:

“So as part of a risk management, they [the client] always want to do it with multiple vendors. When there are multiple vendors it will be a competitive environment, at the same time it will be collaborative because they are developing one piece of work, they are logically splitting the one piece of work by saying vendor 1 you do the development for me, vendor 2 you define specifications for me, vendor 3 you do the testing for me and vendor 4 you do the maintenance for me. So they are splitting the knowledge of this across 4 different vendors so if something happens with one vendor, either the vendor is not behaving well with them or the vendor is becoming too expensive or the vendor is just not able to perform better in which ever condition it is easy for them to transition that work to some other vendor because the knowledge is there, because you split the knowledge across multiple players, and no single vendor will have end to end knowledge and become monopolistic” (Head of North American Banking & Financial Services, Bangalore, January 4th 2012).

The multiple vendor set-up in global software outsourcing relations is not new, and both vendors and clients have worked under such conditions applying waterfall methodology for many years. However, what was unique in this case was that the GSS throughout the years we followed them began to implement agile development in the engagement with the US Bank. At first, the focus was to learn agile development e.g. teaching SCRUM masters etc. At this stage focus was on finding ways to collocate smaller development teams, often at the client site, and run these with agile methodology. However, this type of agile does not scale and the cost of running agile was higher than ordinary projects. Therefore, GSS went through a process by which they developed what they refer to as ‘global agile’, which is a methodology keeping the flexibility in scope and time, while having teams working in globally distributed modes. As, one of the program managers explains:

“the whole shift in the outsourcing model (...) I remember last time when we met (...) I had at least spoken about Agile and Scrum and how it was evolving. (...) Now we have definitely (...) taken a step forward. The distributed model is something which is getting materialized or matured” (Program Manager, Bangalore, January 23, 2013)

In our third round of interviews in Bangalore in 2014, agile development had become the ‘normal’ way of working with the US Bank – and even more interestingly, agile development was implemented not only in a globally distributed mode, it was also combined with a multiple vendor set-up, and it had tremendous impact on the work conditions of the GSS developers and testers.

4.1 From Waterfall to Global Agile

Looking across our interviews, one of the recurrent aspects was the perception of constantly being pushed by the client to produce more and faster, and the interviewees related the higher work pressure to the introduction of the agile methodology in global teamwork with the client. One of the guidelines in agile development is that the work is divided into smaller pieces, each with a clear goal, which can be determined as workable based upon pre-determined criteria. The idea is that the team can work on 'one part' and have this part going through all development phases including testing, and at the end of the development cycle can determine whether they have succeeded in a workable delivery. The development cycles are called 'sprints', and the best practice guidelines for sprints are that they have an interval of between two-three weeks. Initially, when GSS trained their employees in the agile methodology, this was also how they trained and set out guidelines for the initial experiment with global agile. During the years when we conducted interviews with the developers, the intervals of the sprints became still shorter. Not because it was better for the product, for the developers, or for cost saving reasons – but because of a more general pressure from the client to do more in less time, and also to strengthen the client's power position in the multiple vendor set-up.

*"Then they [the clients] also **started rising the bar**, I mean, it is a constant process again, so the more and more I deliver to them, you know, the **expectations keep going up**. So it is not just, you know, at the end of the day it is a process with all, you know, [vendor] companies, not just GSS. And I am pretty sure like, you know, every other [client] company has started to do that. So that is just, you know, **pushing each one of us like**, you know, in thinking through that direction" (Program Manager, Bangalore, February 3rd, 2014, our emphasis)*

What the program manager explains here is how the client pushes the vendor to produce faster and faster by constantly raising the bar of expectations, further increased by the competition in the multiple vendor set-up, where vendors compete at being the best at all times, thus pushing one another. While vendor competition is not new, what is novel is how the competition manifests in the everyday work practices of the developers. Further the delivery manager explains the change from waterfall to agile in the quote below:

*"Okay. In case of scrum, how it works is, the task period is, the length of the task is very small. It will be maximum 1 or 2 weeks. And it will be, their 8 hours work will be tracked. Because we are giving status on daily basis. So if something is assigned today, irrespective of their situation, they have to complete it today. **Whereas in Waterfall, we have a specified target**. Suppose if we need to complete the XX in 2 months, it may happen that, initially, 1 week they may be busy doing something else. They may not be able to spend time with this project. But next week, they will work for 8 hours and try to complete as per the schedule. At least they have that overlapping period bar. Even if they are behind the schedule also, they have time to cope up. **Whereas in Agile, they have to stick to what is being assigned for that specific day**." (Delivery Manager no. 2, Bangalore, February 3rd, 2014, our emphasis)*

What the delivery manager explains is how the shorter sprints (one-two weeks) are organized by tracking eight hours work a day for each developer or tester. This means that the task is split down into smaller tasks, each assigned a number of hours for completion. Each developer is then assigned sub-tasks filling up eight hours a day, and by the end of every day they are providing status. This organizational structure places a strong pressure to complete every day, working constantly for eight hours to achieve the goal, no matter what the situation is, including frequent power outages. If a developer is in a situation when her or his child is getting sick and needs to be picked up from school, the target may not be reached which would make the whole sprint delayed. Add on the constant vendor competition, and the way the actual team involved in the work is a mixture of people from several vendors working on the same sprint driven by the client participation, the expectation to perform in fixed time slots becomes a constant pressure. In the waterfall methodology, developers were able to maneuver and push hours around if needed, however this is not possible in agile, where the tight discipline of the developers is pertinent for the success of the vendor.

Initially it was surprising for us to learn that the IT developers prefer waterfall methodology to agile methodology since the literature on agile methodologies highlights the principles of empowerment of the developer, and how he or she is able to take part in planning the workday and tasks and thereby gets a higher job motivation. Still, what we experienced in our interviews was that the drive for agile methodology did not come from the developers but from the client, and the global agile implementation in combination with the multiple vendor strategy created a most stressful environment for the developers and testers located 'offshore', due to short intervals, tough deadlines and lack of flexibility in planning the work.

4.2 Continuous & Out-of-Sync Work

The competitive environment created by the implementation of global agile also produces uncertain conditions for the private life of the offshore developers in terms of time for vacation or work/life balance on a daily basis. Working on a global scale impacts the work conditions of all people we interviewed. At the higher managerial-level it was clear that the globally distributed work did not allow for longer breaks or vacation. E.g. when asked about his last vacation, the Head of Operations (Bangalore, February 3rd, 2014) told that he had not had any vacation in almost nine years, and in the last year the max time he went away was three days in a row: Friday, Saturday, and Sunday. Other interviewees with top positions also explained how they began work early in the morning and were available for calls from the distant client until 11:00 pm in the evening. While it is not surprising that top managers are required to be flexible at odd times and that the demands are high (this is why they earn the extra money), what did concern us was how the lower level developers and testers also experienced similar demands in flexibility in time and availability – without monetary compensation. According to interview accounts these demands were further increased when agile methodology was implemented in the global collaboration.

A core part of the agile methodology is the daily status meeting, often referred to as the stand-up meeting. Stand-up meetings are short (5-10 min) meetings where the SCRUM

team gathers around the SCRUM board and reports on progress and possible issues. Stand-up meetings are thus a synchronous activity, which in the global setting with 12.5-11.5 hours (depending upon daylight saving) time difference between the client site in Phoenix, Arizona, USA, and one of the vendor sites in Bangalore, India, stretches the flexibility of this vendor's developers. As one of the Delivery managers explains:

“[stand-up meetings] is mostly during Phoenix morning hours and late India hours. So, that way, we have people joining [from the Bangalore] office, some of the people stay back. And some of them if they do not have any work, they go back and they take [the call] from home. (...) It will be around 10'o clock, 10:30pm India time. But our guys also, prefer working a little late because of the traffic we have in India. They do come late. It's their kind of flexibility based on the needs. They come late to office and they stay back. Most of the time, they stay back late and they [take the call] from the office” (Delivery manager no. 2, Bangalore, February 3rd, 2014)

Stand-up meetings executed in the global agile methodology take place in the evening on all weekdays Monday to Friday. This makes it difficult for the Indian developers to take part in any kind of family activity on the evenings during the whole work week. To justify that the developers are staying in the office late, we see in the above quotation that the delivery manager emphasizes that these developers arrive later in the office, and due to the hectic Bangalore traffic this is what they prefer. The delivery manager does not voice any explicit critique of the work conditions and to a certain extent even accepts the working at odd times when phrasing it like: “it's their kind of flexibility based on the [client's] needs”. However, the developers do not really have any flexibility to choose and decide. They are required to stay until late in the evenings at the workplace – a requirement, which did not exist in the waterfall methodology.

When we afterwards explicitly asked the top managers about the developers and testers working in global agile settings, they seemed to be concerned, and they emphasized that it is important for them and the company that the employees are offered good work conditions. However, the constant demands from the client pushing for more as well as the competition among multiple vendors makes it difficult for them to change the work conditions. One high-level manager explained to us that global agile projects were shorter (three-four months), which meant that the highly-structured work was not a permanent condition and thus from his perspective less problematic. However, when we asked a delivery manager (see below) similar questions she explained that the duration of the global agile projects was indeed supposed to be shorter, but that she was currently witnessing a transformation from global agile projects of shorter duration to longer projects:

“Normally projects are four months or five months.... But there are projects, which go for the whole year and then get carried over to the next year. Those are some projects which you know of.” (Interviewer: What do they then do with vacation, if it goes over a whole year?) No, then that is a trouble we have right now. People go under a bit of stress because of this agile methodology. It is happening.” (Delivery manager no. 1, Bangalore, February 3rd, 2014)

The delivery manager is obviously downplaying the work-related stress.. What we see in the above quotation is that while the normal standards for vacation in India is ten fixed dates for public holidays, ten additional days for vacation, and then the possibility for sick leave (this adds up to a total of approximate three weeks in vacation days a year), GSS had difficulties in ensuring that employees get their vacation in the global delivery teams organized through agile methodology. Project periods with the US Bank are extended from four-five months to one year or maybe even into projects of a duration of several years. With the highly-structured day, eight hours a day including daily stand-up meetings every evening 10:30 pm., the global agile methodology tips the work/life balance towards still more work – especially since taking vacation is no longer an option. As the delivery manager further explains:

“During the project, you cannot actually [take vacation]. No. Once the project finishes you go for a party, you go for something to just relax the employees. That is the only thing we try to do. But while the project is on, it is like you have to be on it. You cannot take a leave, it is really stressful” (Delivery manager no. 1, Bangalore, February 3rd, 2014)

It is fine to celebrate a finished project with a kind of party, however it does not make people relax after long periods of heavy work pressure. In our interviews, we listened to vendor managers in GSS, who claimed that the work/life balance of their employees was important for the company, and that, for example, delivery managers tried to offer the best possible work conditions for their developers. However, GSS’s long-term relationship with the US Bank client shapes the middle managers’ maneuverability. Therefore, let us look more closely at the client role in global agile.

4.3 Agile as Global Micromanagement

If the work conditions change for the worse, and you are a company with concerns for your employees, *and* you have good track records with delivery on time and budget following the waterfall methodology, *and* the agile methodology will be costlier for the client – why would you then accept the transformation to agile methodology in global work? We asked several people during the interviews, and unanimously they all said it was because the client wanted it. As a global IT vendor, you need to follow the demands and requirements from the client that outsources its development projects. *“Agile was the trend from the customer”* (Head of Operation, Bangalore, January 23rd, 2013).

Examining our interviews, we detected a transformation of how agile methodology was first introduced and then later became global. Initially, the US Bank wanted to follow the collocation principles of agile, which meant that they wanted GSS to relocate their employees to stay onsite with the client. However, such a set-up is costlier and requires visas for the employees with Indian passports. GSS also recruits locally in near-shore locations (Mexico and Uruguay), but its strength was delivery from teams at various sites in India, among them from the Bangalore area, recruited from the large cohort of well-educated computer engineers. Therefore, the next step for the agile development was to demonstrate and showcase that GSS was able to do agile in a globally distributed manner. In 2014 when we conducted our final interviews the vendor company had managed that.

“We are arriving at that optimal, you know, level where I can still go distributed, and still deliver projects in the agile way” (Program Manager, Bangalore, February 3rd, 2014)

GSS thus had success in demonstrating its business potential for agile development, and it became clear to the client that the methodology did have several benefits such as quicker access to workable deliverables and the possibility to make quick adjustments during the project, and still make the offshore teams deliver on time. The success also made the US Bank interested in engaging in more agile projects, as described below:

“Obviously in the long run we realize lot of benefits come out of the agile process. Business realizes it more because you see things a lot quicker. So, they are more interested in putting projects through agile, so we had an increasing number of agile projects last year” (Delivery manager, Bangalore, February 3rd, 2014)

However, the move towards agile methodology also followed another trend in global outsourcing, namely that the client gets more authority of choosing which specific employees in a vendor company are to work on their projects. One of the ways in which GSS has been able to establish a strong position as vendor is through the company's ability to utilize and further educate graduates in their development projects. An aspect of GSS's delivery model is that each large project would have employees from three different levels of experience and expertise: Top level: more than 8 eight years at work; Middle level: between 3 to 8 years of work experience; and Bottom level: less than 3 years of work experience. For each project, GSS managers would carefully plan as to 'man' the projects and make sure that newcomers were brought in and then also get experience to move up the ladder. In the interviews, we were told that the optimal percentage from each level on a project depends on the type of project as well. It generally is 15-20% top, 60-65% middle, and 20% bottom. However, the clients often demand higher expertise in the project, which means that the vendor may only be allowed to bring in 10% from the bottom level, which makes it increasingly difficult to both train new people and keep the costs low. With the introduction of agile, the clients seemingly get even more authority in deciding upon which employees they want to have in each vendor team. In this way, clients take human resource decisions on behalf of the vendor, directly impacting the vendor's capability to organize and structure a team of specific competencies and skills. Having less bottom-level developers working on clients' projects can thus be counterproductive to growing the vendor organization and its expertise. The clients demand more responsibility and a directing role in the team organization, and then pick and choose individual team members from GSS and the competing vendors. During our interviews, we heard about projects where the area for the delivery was distributed to different vendors, thus the project cut across ten application areas, meaning that the GSS delivery manager had to consult ten vendors during the project (Delivery Manager, Bangalore, February 3rd, 2014). What we see is that with the introduction of global agile, the decision power over the employees working out of India is to a greater extent in the 'hands' of the remote client, in this case located in the US thousands of kilometers away from India and 11.5-12.5 time zones behind India.

The agile methodology, with the detailed task distribution to each team member based upon eight hours of work each day, and the monitoring practice of giving status

updates each evening as individual contributors, makes an excellent tool for the client to micromanaging the IT project at distance. The inspiration for agile methodology was to create autonomous, self-organizing teams, which could take daily decisions upon their own work ensuring excellent systems (Ågerfalk 2006). However, we see that agile mechanisms can also be used by US-based clients to monitor and micromanage the work conditions of IT developers working out of India. We found that the constant push to perform ‘better’, that is faster and at a lower cost, has direct impact upon the conditions for how global agile is performed in practice. The US Bank exercises control over the GSS employees by making them compete against other individuals employed by competing vendors. The US Bank holds all the decision power and bases daily decisions upon individual track records, thus pushing individual employees to achieve high performance all the time.

4.4 Stressful Work Environment

In the globally distributed agile teams, the roles played from offshore (India) are mainly as developers, testers, and analysts – while roles such as SCRUM masters, key architects, and product owners are kept onsite (US). Over the years, it became apparent that employees in GSS working in agile development felt “*more pressure, more time pressure, and stress*” (Program Manager, Bangalore, February 3rd 2014) and that agile “*is very stressful, at the tester level*” (Head of Operation, Bangalore, February 3rd, 2014). While the change in methodology in itself can expose the developers to increased stress, we wanted to explore this issue in more details. Exploring how global agile manifested itself in the everyday work of the so-called ‘techies’, the interviews showed us a pattern whereby global agile in combination with a multiple vendors set-up created certain work conditions for the low-level testers and developers working out of Bangalore. As a delivery manager explained:

“Yes, for the ‘techies’, or for the technical department, it is very stressful, a stressful methodology I would say, because the expectation is too high from the customer’s side. They are expecting every time, once a week, twice a week. The force that I am seeing is, you know, when we started agile, or when we did the certification, it said two to three weeks is a normal, ideal sprint size. But when we started working we have seen a lot of product owners forcing upon us, making it in one week. You know, they are more aggressive, more ambitious. Initially, when they were learning about agile they were not so aggressive, but now I think with smaller projects they have tried out a week’s sprint and now they want to see the deliverables or something after a week, which they call ‘workable’. And that is putting more stress on the people actually.” (Delivery Manager no. 1, Bangalore, February 3rd 2014)

What we see in the above quote is how the US Bank utilizes the transparency and process optimization, which is offered as part of the agile methodology, to monitor and control the outsourced work in detail. The agile methodology stipulates a sprint size of two-three weeks, dependent on the content of the sprint, however the vendor developers offshore experience how the client tends to push for shorter sprints with deliveries within one week. It did concern the Indian delivery managers that the client constantly pushed for faster processes. The stress experienced by the ‘techies’ was rooted in the ways in which

outsourcing IT development was organized between GSS and US Bank. While the client-vendor relationship had existed for decades, interestingly, the justification for moving from the waterfall process model towards the agile process model was not cost saving. Several of our interviewees expressed that agile is not cheaper than waterfall – and that the client will not be able to reduce costs in this way. What we notice was how the agile methodology supported the client in comparing performance measures within GSS's teams, and by using different kind of parameters it also aided the client to compare the performance across the multiple vendor set-up.

“[They have] 15 to 20 parameters every month they look at in terms of, you know, how many projects you delivered, how many projects you delivered on time, on budget, and how many issues reported, how many production back outs, things like that. All those things are captured” (Head of Operation, Bangalore, January 23rd, 2013).

What the above quotation refers to is how all vendors involved in outsourced development projects always compete to deliver on time and budget. All negative issues are reported, which might make the client to swap work from one vendor to another. In addition, it means that competing vendors are directly involved in each project, thus possess the essential knowledge in case the client wants to swap vendor for any sub-project. Moreover, global agile places the client at the center of the development (e.g. in the role as project owner), which means that not only is it possible to ‘easily’ swap vendor, in addition the client has direct access to the human resources in the project and can be proactive in taking decisions upon task distribution on the day-to-day level. Global agile in multiple vendor set-ups might not cut costs, but it does provide direct access to knowledge and makes the client less dependent upon one vendor by utilizing ‘divide and conquer’ strategies across multiple vendors.

“So they [the clients] felt they wanted that flexibility in terms of having the knowledge..., rather than just, you know, being depending on the vendor, that was one key transformation.” (Program Manager, Bangalore, February 3rd, 2014).

The transformation towards global agile was thus driven by the client's interests in greater flexibility and maneuverability combined with access to highly qualified knowledge from the selected strong IT vendors that can drive the client's business and technology development. Further, global agile solves many of the issues often found in large IT projects, by supporting closely coupled work across geography and time (Jensen 2014), supporting common ground across organizations (Olson and Olson 2014), and commitment to tasks and deliveries (Mark 2002). However, the cost of this scenario is that the every-day work conditions of the low-level testers and developers working at sites at great distance of the client are worsened.

5. DISCUSSION

Waterfall methodologies for structuring IT development through rigorously sequential processes, following principles of top-down, one-way phases have been linked to micromanagement or scientific management, as initially developed by Ford's mass productions of cars (Havn 1991, Nicholson and Sahay 2001, Parthasarathy 2004). Agile

development has been seen as a response to the criticisms and as a move away from highly structured processes, regaining the emphasis on the developers involved, their learning, and their decision-making processes. A misconception of agile development is that it requires no processes, documentation, or planning, and thereby risks producing low quality work and badly designed software systems (Janes and Succi 2012). However, our empirical study unveiled an opposite risk. Namely how an over-commitment to processes, plans, and tools combined with agile methodology in a global outsourcing setting risks losing sight of the individual circumstances under which IT developers offshore perform their daily work.

In our case the implementation of global agile was based upon the demand from a major Western client. In practice, it became a mechanism to improve productivity through formalization and division of software tasks into small sub-tasks, which offshore IT developers were pushed to finish through continuous shortening of the deadlines and constant control orchestrated by a remote client through its competitive multiple vendor set-up. Further, the technological and organizational separation created by this multiple vendor set-up spurred competition locally (within India across IT vendors) to the benefit of the remote client. By making IT vendors with teams located in India compete against each other in terms of price and time, these contractual engagements directly affect the work of the IT developers in their transnational practices. When the client is in charge of the decision of which vendor should work on certain parts of the technology, and the client can change this decision any day based upon the continuous monitoring of performances, the client has the power and control, which directly impact the work of the IT developers.

The performance of global agile thus becomes a way of disciplining the IT developers in their individual work, producing a certain logic of transnational encounters, resembling a post-colonial variant of the 'divide and rule' strategy (Mayasandra, Pan et al. 2006). When the Indian IT developers produce transparent work processes as stipulated by the agile methodology and supported by the tools and technologies, it becomes possible for the remote Western client to engage in what Shome refers to as practices of gendered surveillance produced by digital capitalism (Shome 2006). Here 'gendered' does not refer to the dichotomy of female and males, but instead it refers to unequal relationships, where mutual respect and acknowledgment are *not* co-exchanged, but instead are performed as a sub-ordinary unbalanced association between the client and the vendor. Thus, global agile assumes that IT development is similar to industrial production and that the IT developer is a component of that production, which can be replaced easily. This assumption about the nature of software development has been strongly disputed by several computer science experts since the 1980s, e.g. the Danish Turing award winner Peter Naur (Naur 1985, p. 260). His argument is that software development is a creative profession, which requires expertise and knowledge in other ways than industrial production. Software development is not fitted for the assembly line. Nevertheless, the client's assumptions about software development as a kind of assembly line work is evident in our empirical material.

The pressure and transparency produced by the agile algorithm meant that offshore IT developers had increased stress level and less agency in organizing their own work

compared to those using waterfall models. The waterfall model provided ways for the IT developers to organize and align their work practices also dealing with the continued unpredictability in establishing the exact time it takes to finish a programming task and make ‘workable deliverables’. Furthermore, the algorithmic machine also affected the organization of the IT developers’ private lives. While we encountered several high-ranking female managers in our interviews, we also heard stories about how female engineers at lower echelons left the company when they were married, since it was difficult to combine career and family life. While such stories are not unique to India, it is important to notice that the increased pressure in work introduced by agile in the global outsourcing set-up risks pushing more female IT developers out of the workplace, since the flexibility required for making work and family life co-exist is reduced. In another study of global software development in Bangalore, it was found that in particular female low-level developers and testers often live together in women’s hostels close to the work place during the week and then ‘migrate’ back to their families during the weekend (Matthiesen and Bjørn 2016). While we were not able to explore housing conditions in our case study, it is evident that any IT developer, who lives far away from the office and works under global agile outsourcing conditions is required to stay back in the office on late evenings. Working long hours and postponing vacation could potentially be self-imposed, meaning that the IT developers in Bangalore feel a strong loyalty to the company and the project, as well as a responsibility to stay late when it is needed. However, our empirical material shows that it is not that simple. In our interviews, the majority of the team leads, and project leads, all had the expectation that their work would interfere with their private lives. When being in ‘managerial’ positions, even at a lower level, it is perceived as necessary and important to work extra hours to make the project run smoothly. However, when we address the lower level IT developers and testers (“the techies”), the pressure to stay late is not self-imposed. Instead staying late is a feature of their work conditioned through the multiple vendor set-up of global agile with remote foreign clients setting the agenda and working out of different time zones.

5.1 Global Outsourcing in Transnational spaces

Working in *transnational spaces* based upon global outsourcing set-ups makes the relationship between the ‘onsite’ and ‘offshore’ important to scrutinize. In our interviews, all the IT developers working out of India refer to their own physical location in Bangalore as ‘offshore’, thereby taking the client’s perspective. Experiencing and naming the location of one’s body as ‘offshore’ creates a certain logic as to what becomes ‘center’ and what becomes ‘periphery’. At any time, the IT developers paid attention to ‘onsite’ decision makers, and accommodated their convenience by working during odd hours which fit ‘onsite’ employees. Interestingly, the vocabulary used was that the vendor company’s IT developers do work ‘*offshore*’ even though they are at ‘home’ in India.

The majority of the people involved in global software outsourcing are geographically located in what have been referred to as the Global South (Walsham 2008) e.g. India, China or the Philippines. This means that we might consider the Global South the center of the global work. We might explain this empirical observation in terms of power relations between headquarters and subsidiaries (Hinds, Retelny et al. 2015), between client and vendors (Søderberg and Romani 2017). Our case demonstrates an asymmetric

power relation between the remote Western client and the competing vendor organizations located in India. When we explore the ways in which this asymmetric relationship becomes enacted in the performance of the work (Vlaar, Febema et al. 2008), we find that examining the translocality of the experienced situation of global work from the perspective of the low level IT developers and testers in Bangalore, the discursive performance of 'home' becomes 'away' from the center, which is located 'onsite'. It becomes clear that the ordinary use of words like 'home' and 'away' gets transformed when adding the translocality lens to the analysis of global outsourcing. The power and decision-making are unequally distributed between the discursive and bodily manifestations of 'away'/'offshore' and 'home'/'onsite'. Knowledge and power play out in the specific work practices and are fundamentally based upon the distinction between low status and high status in work (Matthiesen and Bjørn 2017). Who owns the code is an important question to ask in global outsourcing, when we explore power. It can be determined by asking who are responsible for the high status work such as designing the software system, and who are responsible for the low status work of executing pre-defined tasks and low-level coding (Metiu 2006). While GSS does not only do low-level coding, the ways global agile was performed in this case was connected determined by the ownership and knowledge of the code-base.

The power division strategies in our empirical material related to owning the code took a specific form. When the client decides to separate the technical development across multiple vendors, it means that each vendor only has a piece of the complete system, thus the collective technical knowledge only resides with the client. Only specific aspects of technology strategies will be shared and discussed with individual IT vendors, but core knowledge about the infrastructures in the system rests with the client who has the full knowledge and insight into the system. Clients might choose to change technological strategies directly impacting the everyday work conditions for IT developers in the vendor organization. The client is part of the team (playing the role as the project owner), thus the power to take decisions still remains in the team following the recommendations related to agile methodology. However, differently from the original agile agenda, where a core element is that teams, if not organized in completely horizontal and symmetric relationships, at least experience themselves as self-organized and self-managed teams; the agile teams in global outsourcing were characterized by clear asymmetric relations, and IT developers in India did not experience that they had control over their own work. While asymmetry always exists in client/vendor relationships, what makes the global agile particularly problematic is how the agile methodology provides the client an opportunity to constantly control and evaluate the offshore team members' performance, competencies, and qualifications from a remote location, through the production of transparent processes supported by tools and technology.

A translocal lens on global outsourcing thus directs special attention towards what is made visible and what remains invisible for whom in the transnational work practices. Our empirical study demonstrates that *what is made visible to the remote client* is the low-level testers' and developers' performance, progress, and specific competences produced by the agile methodology. In this way the global agile methodology in our outsourcing setup resembles body shopping at a distance (Bhatnagar and Madon 1997, Parthasarathy 2004), since remote clients can pick and choose team members based upon

continuous evaluation of their individual performances. Thus, it becomes the daily responsibility of the individual employee to ensure that their roles and tasks are not redistributed to team members in competing vendor companies. *What remains invisible* are the everyday work conditions of working ‘away’ while at ‘home’, of working ‘offshore’ controlled by ‘onsite’ created by the transnational work set-up, highly impacting the work as well as the lives of the IT developers.

Previous research points to how workers in outsourced projects experience contradictions between on the one hand having the privilege of becoming part of the global information economy, while simultaneously being subject to temporal complexities (Poster 2007, p. 57). Our study shows that IT developers working under the time regime of global agile experienced the temporal management of increasingly short deadlines by constantly reduced sprint periods, daily stand-up meetings, as well as the micro-coordination of clock time devoted to particular sub-tasks and the precise calculations to optimize their work. The global agile changed the temporal ordering of work in other ways than the waterfall methodology. The IT developers working in global agile, thus shifted their work time to be able to participate in the daily stand-up meeting in the late evening, Monday to Friday. Working in transnational time applying the agile methodology creates a complex production of *in-betweenness*, where participants concurrently work and live across multiple nations and geographies, which intersect in the collision of multiple local times (Shome 2006). Global agile forced the ‘techies’ to work out-of-sync locally, in order to be in-sync remotely. Being out-of-sync locally risks impacting the personal lives of the IT developers in terms of preventing them from participating in social activities in their local environment during the evenings.

5.2 Translocality in Globally Distributed Work

In our empirical study, we have investigated a particular type of distributed work, namely global IT outsourcing applying agile methodologies in a multiple vendor setup. The context of our study was thus shaped by the particularities that the empirical material provided, which lead to our findings concerning the risks and problems when introducing agile into globally distributed work. However, if we broaden our scope, we propose that translocality as an analytical approach can also help us to unpack distributed work in other work situations. What does it mean to study translocality in distributed work?

When we study translocality in distributed work, the boundaries for the object of study are not pre-determined. Instead the object of study is emergent and shaped by topics and themes, which appear during the study. Therefore, we should be careful not to exclude important connections and associations prematurely in our investigations, since we risk missing important insights, e.g. stressed work life turned out to be an important topic for us, when examining the implementation of global agile. This means that the brackets for what is inside or outside of the research scope is not pre-given or pre-existing. Instead the scope is created, transformed, or perhaps collapsed depending upon how the study evolves.

In studies of translocality in distributed work topics such as occupational politics, infrastructural connectivity, and international agency become essential. In our case study,

occupational politics, as in work time, leisure, and vacation, was not an issue, which the IT developers or the vendor company could decide upon independently. Instead, it was dependent upon the client's choice to engage in micromanagement at a distance requiring certain coordination events to happen every evening five days a week. Or it was the client's decision that projects might extend over several months or even more than a full year, meaning that IT developers hardly did not get any breaks or vacation days. The *infrastructural connectivity* in our case emerged in different ways. Firstly, we experienced that IT developers and testers did their daily SCRUM meeting at the office. Previous research in other IT companies in Bangalore reported that low-level IT developers and testers do not have access to laptops and cannot work from home due to security issues (Matthiesen and Bjørn 2016). Also, in our case study, the IT developers did not have access to homework stations and laptops on a regular basis. Not having access to work from different locations (home and in the office) is thus part of the infrastructural connectivity. By only being able to connect to the 'global work' from the office sets special conditions for the IT developers' work.

Secondly, infrastructural connectivity is also related to the ways in which the local infrastructures support the IT developers in *getting connected* to the global work. Transportation in Bangalore for local IT developers has been explored elsewhere (e.g. D'Mello 2005, D'Mello and Sahay 2007) and anybody who has tried to cross the city during rush hours knows the hassle and long time this transportation takes. However, if connectivity to the global work only happens at the office, it also means that travelling locally to the office is part of the infrastructural connectivity relevant for consideration.

When it comes to *international agency*, as in the capacity to act on the scene of global work, it is evident that GSS as a large multinational company is extremely successful and also has clients all over the world. However, if we zoom in on the low-level IT developers and testers, the foundation for the concrete execution of global agile, we find that international agency becomes the vendor employees' capability to constantly adapt to increasingly short deadlines, to work out-of-sync locally, and, to adjust to decisions about the work tasks taken by remote clients. International agency for the offshore IT developers is a way to increase opportunities for getting interesting tasks and to seize new learning by improving skills and capabilities, but with agile it also introduces agendas of human-automation based upon relational asymmetry.

When we bring in translocality in our study of distributed collaboration, we pay attention to differentiations, which emerge in the empirical case. We saw how the differentiation between vendors in India working together with the same remote client was not a viable 'grouping' to understand the challenges of global agile. Despite of being in competition – all the vendors working out of Bangalore were actually part of the same groupings (as in offshore team members) when the US client entered the everyday work practices. Certainly, the introduction of the multiple vendor perspective created some of the pressure on the IT developers, however such tensions also existed in the waterfall methodology. Leaving the 'fixed boundaries' and groupings, we understand that it is not simply client versus vendor – but the ways in which a remote Western client's behavior impact the work conditions for Indian employees across multiple vendor companies.

What was important in understanding the impact of global agile was how the competition was intensified with the introduction of global agile not only for the vendor company at large, but as an individual experience in the everyday work. The lack of boundaries for work was evident in the ways in which daily schedules were handled and reported upon, as well as in the continuous flow of work and lack of breaks. Technologies play an important role for how boundaries are enacted, and in our case in the ways in which visibility is accomplished in a one-way manner. It was the visibility of the Indian IT developers' status for the day, which made it possible for the US client to monitor and act accordingly by adding more pressure or reallocating work between vendors. However, visibility was not mutual, thus decisions might be taken by the client without the IT developers understanding the client's perspective and situation.

We propose three recommendations for using translocality to study distributed work, which we believe can guide future research.

Recommendation no. 1: Translocality in distributed work must consider the experienced localized situations taking into account the occupational politics, infrastructural connectivity, and international agency

Recommendation no. 2: Translocality in distributed work refutes fixed or pre-defined boundaries as explanations, and instead insists on bringing attention to emergent types of belonging, social grouping, and differentiation in practice

Recommendation no. 3: Translocality in distributed work pays attention to the ways in which tools and technologies are enacted in transnational practice and produce certain logics about time and space.

6. CONCLUSION

The word 'global' in globally distributed work often entails the assumption that it is possible to talk about and refer to something 'outside the local practice'. That a difference exists between what is 'global' and what is 'local'. Differently, translocality repudiates the dichotomy of global/local as a useful analytical lens, and insists that nothing can be understood as a 'place from nowhere', but instead is always understood from the immediately localized and situated of an experienced practice. This means that we cannot determine the impact of global agile as a methodology independent of how it unfolds at particular sites. Translocality guides us to explore global agile as a localized experience from the offshore IT developers' perspective working out of Indian sites, while being connected to and associated with the client representatives in the US. These associations are dynamically enacted through the tools and technologies by which the IT developers engage with remote colleagues through digital connectivity. Nevertheless, the associations are always experienced locally. The associations across sites do not get disconnected, when the synchronous activities end and video conferencing equipment is shut down. It is not *only during* the stand-up meetings in the late evening in Bangalore that the agile methodology impacts the IT developers' work conditions. Instead the associations continue to exist and regulate the daily time schedules controlling the organization of globally distributed work. The concept of translocality makes us notice

and include additional details on how the micromanagement of the individual workday not only impacts the scheduling of the work, but also personal life outside the workplace. The concept of translocality guides us to consider the experienced work arrangements as a multiplicity of occupational politics, infrastructural connectivity, and international agency, which reach beyond national borders. The IT developers located in India work under different transnational circumstances than IT developers located in the US or Europe. Understanding their lived work experiences in transnational work thus requires us to pay attention to the ways in which methodology, technology, and work organization shape the translocality of the workplace in certain ways.

Vocabulary does matter (Cohn 1987); thus by expanding current HCI vocabulary on globally distributed work bringing in translocality as a lens, we manage to illuminate and detect aspects, which formerly have been neglected. Thus, our paper introduces translocality in globally distributed work, and demonstrates how the dark side of global agile methodology arises as a surprising outcome of such an analysis and thereby pushes the research of global software outsourcing forward.

ACKNOWLEDGEMENT

The argument in this paper has benefited from many of the discussions, we have had with researchers concerning the difficulties talking about issues of politics in global software outsourcing. We want to thank the Indian IT service provider, which made this study possible, by inviting us to interview their employees over a period between December 2011 and February 2014. We appreciate all the time people have provided us to discuss these issues at events such as the ISTC all-hands event in 2014 and the Dagstuhl seminar Rethinking Software Productivity in 2017. We would like to acknowledge the great feedback and discussions we have had with several scholars about the issues discussed in this paper including James Herbsleb, Susanne Bødker, Gloria Mark, Morten Kyng, Lone Koefoed Hansen, Jack O'Neil, Pamela Hinds, William Scherlis, Christian Kästner, Stina Matthiasen, Irina Shklovski, Paul Dourish, Mel Gregg, Silvia Lindtner, Katie Pine, Marisa Cohn, and Manju Ahuja. Finally, we would like to thank the critical and engaging feedback from the Associate Editor and anonymous reviewers, which helped us developing our argument. The empirical study was conducted within the project "Next Generation Technologies and Processes for Global Software Development," financially supported by the National Council for Strategic Research, Ministry of Science, Innovation and Higher Education in Denmark (Grant no. #10-092313).

REFERENCES

- Ågerfalk, P. (2006). Towards better understanding of agile values in global software development. Modeling methods for systems analysis and design, EMMSAD: Via Nova Architectura.
- Bhatnagar, S. and S. Madon (1997). "The Indian software industry: moving towards maturity." Journal of Information Technology **12**: 277-288.

- Bjørn, P. (2003). Re-Negotiating Protocols: A way to Integrate Groupware in Collaborative Learning Settings. ECIS, New Paradigms in Organizations, Markets and Society, Proceedings of the 11th European Conference on Information System, Napoli
- Bjørn, P. (2016). "New fundamentals for CSCW research: From distance to politics." Interactions(May-June): 50-53.
- Bjørn, P. and N. Boulus-Rødje (2015). "The multiple intersecting sites of design in CSCW research." Computer Supported Cooperative Work (CSCW): An International Journal **24**(3): 319-351.
- Bjørn, P. and L. R. Christensen (2011). Relation work: Creating socio-technical connections in global engineering. European Conference on Computer supported cooperative work (ECSCW). Aarhus, Denmark, Kluwer Academic: 133-152.
- Bjørn, P., M. Esbensen, R. E. Jensen and S. Matthiesen (2014). "Does distance still matter? Revisiting the CSCW fundamentals on distributed collaboration." ACM Transaction Computer Human Interaction (ToChi) **21**(5): 1-27.
- Bjørn, P. and M. Hertzum (2006). Project-Based Collaborative Learning: Negotiating Leadership and Commitment in Virtual Teams. 5th Conference on Human Computer Interaction in Southern Africa (CHI-SA), Cape Town, South Africa, ACM SIGCHI.
- Bjørn, P. and O. Ngwenyama (2009). "Virtual Team Collaboration: Building Shared Meaning, Resolving Breakdowns and Creating Translucence." Information Systems Journal **19**(3): 227-253.
- Bjørn, P. and C. Østerlund (2014). Sociomaterial-Design: Bounding technologies in practice, Springer.
- Blomberg, J. and H. Karasti (2013). "Reflections on 25 years of ethnography in CSCW." Computer Supported Cooperative Work (CSCW): An International Journal **22**(4-6): 373-423.
- Boden, A., G. Avram, L. Bannon and V. Wulf (2009). Knowledge management in distributed software development teams: Does culture matter? International conference on Global Software Engineering (ICGSE). Limerick, Ireland, IEEE Press: 18-27.
- Boden, A., B. Nett and V. Wulf (2007). Coordination practices in distributed software development of small enterprises. International Conference on Global Software Engineering (ICGSE). Munich, Germany, IEEE Press: 235-244.
- Christensen, L. and P. Bjørn (2014). Documentscape: Intertextuality, sequentiality and autonomy at work. ACM CHI Conference on Human Factors in Computing Systems Toronto, ON, Canada, ACM.

- Cohn, C. (1987). "Sex and death in the rational world of defense intellectuals." Signs: Within and without: Women, gender, and theory **12**(4): 678-718.
- D'Mello, M. (2005). "Thinking local, acting global: Issues of identify and related tensions in global software organizations in India." Electronic journal of information systems in developing countries **22**(2): 1-20.
- D'Mello, M. and S. Sahay (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." Information and Organization(17): 162-192.
- Esbensen, M. and P. Bjørn (2014). Routine and standardization in Global software development. GROUP. Sanible Island, Florida, USA, ACM.
- Esbensen, M., P. Tell, J. Cholewa, M. Pedersen and J. Bardram (2015). The dBoard: A digital scrum board for distributed software development. ITS. Funchal/Madeira, Portugal, ACM: 161-170.
- Gertsen and A.-M. Sørenberg (2012). Global collaboration: Intercultural experiences and learning. New York, USA, Palgrave Macmillan.
- Grinter, R. E., J. D. Herbsleb and D. E. Perry (1999). The Geography of Coordination: Dealing with distance in R&D Work. GROUP, Phoenix, Arizona USA, ACM.
- Havn, E. (1991). "In serach of new structures: Issues in designing integrated manufacturing systems." International Journal of Human Factors in Manufacturing **1**(4): 339-350.
- Herbsleb, J. (2007). Global software engineering: The future of socio-technical coordination. Future of Software Engineering (FOSE). Washington, DC, USA, IEEE Computer Society.
- Herbsleb, J. and A. Mockus (2003). "An empirical study of speed and communication in globally-distributed software development." IEEE Transactions on Software Engineering **29**(3): 1-14.
- Hinds, P. and M. 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.
- Hinds, P., T. Neeley and C. Cramton (2014). "Language as a lighting rod: Power contests, emotion regulation, and subgroup dynamics in global teams." Journal of international business **45**(5): 536-561.
- Hinds, P., D. Retelny and C. Cramton (2015). In the flow, being heard, and having opportunities: Sources of power and power dynamics in global teams. Computer Supported Cooperative Work (CSCW). Vancouver, BC, Canada, ACM: 864-875.

- Hinds, P. J. and D. E. Bailey (2003). "Out of Sight, Out of Sync: Understanding Conflict in Distributed Teams." Organization science **14**(6): 615-632.
- Janes, A. and G. Succi (2012). The dark side of agile software development. Onward! Tucson, Arizona, USA, ACM.
- Jensen, R. E. (2014). Why closely coupled work matters in global software development. GROUP. Sanible Island, Florida, USA, ACM.
- Jensen, R. E. and P. Bjørn (2012). Divergence and convergence in global software development: Cultural complexities as societal structures. COOP: Design of cooperative systems. France, Springer: 123-136.
- Jensen, R. E. and B. Nardi (2014). The rhetoric of culture as an act of closure in cross-national software development department. European Conference of Information System (ECIS). Tel Aviv, AIS.
- Levina, N. and E. Vaast (2008). "Innovating or doing as told? Status differences and overlapping boundaries in offshore collaboration." MIS Quarterly **23**(2): 307-332.
- Marcus, G. (1998). Ethnography through thick and thin. New Jersey, Princeton University Press.
- Mark, G. (2002). "Conventions and Commitment in Distributed CSCW Groups." Computer Supported Cooperative Work (CSCW): An International Journal **11**: 349-387.
- Matthiesen, S. and P. Bjørn (2015). Why replacing legacy systems is so hard in global software development: An information infrastructure perspective. CSCW. Vancouver, Canada, ACM.
- Matthiesen, S. and P. Bjørn (2016). Lets look outside the office: Analytical lens unpacking collaborative relationships in global work. COOP2016. Trento, Italy, Springer.
- Matthiesen, S. and P. Bjørn (2017). "When distribution of tasks and skills are fundamentally problematic: A failure story from global software outsourcing." PACM on Human-Computer Interaction: Online first 2018 ACM Conference on Computer-supported Cooperative Work and Social Computing **1**(2, Article 74): 16.
- Matthiesen, S., P. Bjørn and L. M. Petersen (2014). "Figure Out How to Code with the Hands of Others": Recognizing Cultural Blind Spots in Global Software Development. Computer Supported Cooperative Work (CSCW). Baltimore, USA, ACM.
- Mayasandra, R., S. Pan and M. Myers (2006). Viewing information technology outsourcing organizations through a postcolonial lens. IFIP Social inclusion:

- Societal and organizational implications for information systems. E. Trauth, D. Howcroft, T. Butler, B. Fitzgerald and B. DeGross, Springer. **208**: 381-396.
- Metiu, A. (2006). "Owning the code: Status closure in distributed groups." Organization science **17**(4): 418-435.
- Naur, P. (1985). "Programming as theory building." Microprocessing and microprogramming **15**: 253-261.
- Nicholson, B. and S. Sahay (2001). "Some political and cultural issues in the globalisation of software development: Case experience from Britain and India." Information and Organization **11**: 25-43.
- Olson, G. M. and J. S. Olson (2000). "Distance Matters." Human-Computer Interaction **15**: 139-178.
- Olson, J. and G. Olson (2014). Working together apart: Collaboration over the internet, Morgan & Claypool Publishers.
- Parthasarathy, B. (2004). "India's silicon valley or silicon valley's India? Socially embedding the computer software industry in Bangalore." International Journal of Urban and regional Research **28**(3): 664-685.
- Philip, K., L. Irani and P. Dourish (2012). "Postcolonial computing: A tactical survey." Science technology & Human Values **37**(2): 3-29.
- Poster, W. (2007). "Saying 'good morning' in the night: The reversal of work time in global ICT service work." Workplace Temporalities: Research in the Sociology of Work **17**: 55-112.
- Ravishankar, M. N., S. L. Pan and M. D. Myers (2013). "Information technology offshoring in India: A postcolonial perspective." European Journal of Information Systems **22**: 387-402.
- Romani, L., S. Sackmann and H. Primecz (2011). Culture and negotiated meaning: The value of considering meaning systems and power imbalance for cross-cultural management. Cross-cultural management in practice: Culture and negotiated meanings. H. Primecz, L. Romani and S. Sackmann. Cheltenham, Edward Elgar: 1-17.
- Shklovski, I., J. Vertesi and S. Lindtner (2014). "Introduction to this special issue on Transnational HCI." Human-Computer Interaction **29**(1): 1-21.
- Shome, R. (2006). "Thinking through the diaspora: Call centers, India, and a new politics of hybridity." International journal of cultural studies **9**: 105-124.
- Søderberg, A.-M. (2015). Indian boundary spanners in cross-cultural and inter-organizational teamwork: An account from a global software development

- project. Routledge Companion to cross-cultural management. N. Holden, Tietze and Micheaelova. London, Routledge: 334-343.
- Søderberg, A.-M. and N. Holden (2002). "Rethinking cross cultural management in a globalized business world." International Journal of Cross Cultural Management (CCM) **2**(1): 103-121.
- Søderberg, A.-M., S. Krishna and P. Bjørn (2013). "Global Software Development: Commitment, Trust and Cultural Sensitivity in Strategic Partnerships." Journal of International Management **19**(4): 347-361.
- Søderberg, A.-M. and L. Romani (2017). "Boundary-spanners in global partnerships: A case study of an Indian vendor's collaboration with Western clients." Group & Organization Management **42**(2): 237-278.
- Souza, C. d. and D. Redmiles (2007). The awareness network: To whom should I display my action? And, whose actions should I monitor. European Conference on Computer Supported Cooperative Work (ECSCW). Limerick, Ireland, Kluwer Academic: 99-118.
- Subramanian, R. (2006). "India and information technology: A historical and critical perspective." Journal of Global Information Technology Management **9**(4): 28.
- Tøth, T. (2015). Trustworthiness: Enabling global collaboration: An ethnographic study of trust, distance, control, culture and boundary spanning within offshore outsourcing of IT services. PhD Series 03.2015, Copenhagen Business School, CBS.
- Vlaar, P., P. Febema and V. Tiwari (2008). "Cocreating understanding and value in distributed work: How members of onsite and offshore vendor teams give, make, demand, and break sense." MIS Quarterly **32**(2): 227-255.
- Walsham, G. (2008). ICTs and global working in a non-flat world. IFIP: Information technology in the service economy: Challenges and possibilities. Boston, Springer: 12-25.
- Watson-Manheim, M. B., K. Chudoba and K. Crowston (2002). "Discontinuities and Continuities: A New Way to Understand Virtual Work." Information Technology & People **15**(3): 191-209.
- Williams, A., S. Lindtner, K. Anderson and P. Dourish (2014). "Multisited design: An analytical lens for Transnational HCI." Human-computer interaction **29**(1): 78-108.