

# New Fundamentals for CSCW Research: From Distance to Politics

---

## Insights

- Collaboration readiness is a fundamental challenge in global software development.
- A key design challenge in distance collaboration is supporting the extra work required to create sociotechnical connections through stable, accessible, available technologies.
- Collaboration across distance is best done in closely coupled work-task situations.

Fifteen years ago a famous paper by Gary and Judith Olson taught us that distance matters in collaborative settings [1]. Back then, the technologies supporting collaboration between geographically distributed employees within organizations were still a rather new phenomenon, and technologies were measured by their ability to mimic collocation, which was seen as the optimal way to collaborate. Multiple studies demonstrated how important interactions (e.g., awareness and coordination activities) were difficult to achieve in a distributed manner; seamless interaction required articulation work, which was challenging in a geographically distributed setup. Basically, in 2000,

collaboration across distance was seen as *unlikely* to succeed, except in situations of *high common ground*, *readiness for collaboration*, and *readiness for collaborative technologies*, in a *loosely coupled setting*, where it had a *chance* to succeed [1]. This means that the distance framework had four main attributes: common ground, collaboration readiness, collaboration technology readiness, and coupling of work.

Today, collaboration across distance is the new normal for many knowledge-heavy professions such as engineering, research, and software development. When you walk into any software engineering company, you will be met with teams whose



participants are located in offices spread around the world who are using collaborative technologies that are seamlessly embedded into their everyday lives. In light of these new practical realities for distance collaboration, we found it interesting to explore whether the CSCW fundamentals needed to be revisited empirically and theoretically [2].

*NexGSD* is an interdisciplinary research project that ethnographically studies the collaboration activities in several global software development companies. Our aim is to design technologies to support such collaboration. We are a team of academics collaborating with industry partners. Over the past couple of years,

we have conducted several studies of global software development where developers are located in Denmark, India, the Philippines, Germany, the U.K, and the U.S (see sidebar). We decided to interrogate the CSCW fundamentals by challenging the framework with four empirical studies. These studies were diverse in nature, but all demonstrated collaboration across geographical distance and were therefore appropriate. We explored these cases by examining the empirical data of each as it relates to common ground, collaboration readiness, collaboration technology readiness, and coupling of work.

**Common ground.** All our cases demonstrated that common ground

continues to be a fundamental challenge in distributed collaboration. It is still a challenge for collaborative partners to share a common vocabulary and interpretations of interaction, and to actually know they have these shared practices in common. However, our findings also show how we might further nuance the challenge of common ground, particularly in cases of global software development. Namely, by distinguishing common-ground challenges into 1) the practices of creating and developing shared language and specific professional domain knowledge, and 2) the practices of identifying, creating, maintaining, and applying processes and methods that support global collaboration.

## FACTS ABOUT CASES

- **GLOBALSOFT** is a study of the impact on collaboration when an outsourcing setup transforms into an offshore setup of global software development between a Danish company and a company located in the Philippines [3]. The study took place over a three-year period, during which we spent time observing in both Denmark and the Philippines, following the daily work in a challenging project with the aim of designing a new system for Danish society.
- **INIT** is a study of global software development practices from an Indian vendor perspective. It was a three-year study in which we followed the transformation of three large-scale IT-outsourcing projects with American and European clients [4]. Each year we traveled to Bangalore, India, to interview the participants in the projects, focusing on conditions for collaborative practice, which changed over the study period. We also visited locations around Europe.
- **SCANDIABANK** is a study of the changes in work practices in Denmark and in India when the company initiated global outsourcing relationships [5]. It was a six-month study in which ethnographic observations were done in both the Danish and Indian offices for longer periods.
- **INDK** is a focused study of the dependencies in global Scrum work organizations in teams dispersed geographically between Denmark and India [6]. The study consisted of two Ph.D. students being physically located at the same time at each location for a period of three weeks. During these weeks, the software team was observed in all their meetings and interactions; in particular we explored the interactions in their daily Scrum meetings.

### *Collaboration readiness.*

Collaboration readiness also continues to be a fundamental challenge in collaboration among geographically distributed partners. In line with others, our empirical cases pointed to how assessing collaboration readiness should be explored not only in terms of geographical distance, but also in terms of the diverse set of discontinuities that exist in each particular case, including time, culture, professions, technologies, methods, and work practices. In addition, we found that assessing collaboration readiness cannot be done in terms of one single continuum between high and low. Instead, collaboration readiness involves a multiplicity of dimensions and relations where collaborative partners can be at different stages at the same time for some dimensions, and at the same stage at different times for other dimensions.

***Collaboration technology readiness.*** All our cases of distributed collaboration took place in software

development settings, which means there might be a bias here. However, we found that collaboration technology readiness, conceptualized as “knowledge about new technological opportunities and willingness to adopt,” was not an issue in any of our cases. All participants insisted that it was not additional technologies they were lacking, since they had plenty. At first, this was of course troubling for us, since our aim was to design new tools for the setting. However, we found that it was *not* because the software developers had *no* challenges related to technologies—our observations pointed to how they clearly had. But the nature of the challenges could not be captured by the term *collaboration technology readiness*. Instead, our findings suggest that future research should investigate technological appropriation in terms of the extra work required to create sociotechnical connectivity in distributed work—for example, relation work—and that the design challenges relate to designing for

stability, accessibility, and availability of collaborative technologies in distributed work.

***Coupling of work.*** The final attribute from the original distance framework is the preference for loosely coupled work arrangements. Interestingly, our case demonstrated quite the opposite. In all our cases, the situations where the work task created closely coupled engagements between participants, despite their geographical dislocation, were the most optimal. In fact, in all our cases the groups were adopting new work methodologies such as Agile (Scrum, etc.) during the period in which we engaged with our field sites. This of course made us puzzle about why, here, 15 years later, loosely coupled work cannot be identified as a prerequisite for successful collaboration in distributed work.

The simple answer is: Because software developers are lazy. In general, software developers try to make their work tasks as simple as possible, which ideally means fixing things themselves, without too much overhead from working with others. Each time we have a collaborative situation, we have work and articulation work, where articulation work is all the extra work required to make the work function. In distributed collaboration, the efforts involved in handling articulation work increase, since dedicated planned interaction through technology does not happen automatically just because a technology has been introduced. Instead, software developers need to spend time and effort in making the technology function—for example, setting up video-meeting rooms—and often they experience the technology as unstable, unreliable, and requiring of additional effort on top of current difficulties in collaborating with their remote colleagues. This means that if their work task does not force them to interact with remote partners, they will not do it. In situations of closely coupled work, the work task requires the remote collaborators to interact, which means they will be willing to spend the time and effort involved in engaging in technology-mediated collaborative activities. In all our cases, situations of closely coupled work made the distributed software developers engage in frequent interaction through technologies—they were forced

---

**In general, software developers try to make their work tasks as simple as possible, which ideally means fixing things themselves, without too much overhead from working with others.**

to have frequent interaction. These were the situations where participants felt that the collaboration across geographical distance worked best. While previous work has pointed to the importance of frequent interaction, our point here is different, namely, that it is the nature of the task (the closely coupled task) that makes software developers willing to spend time and effort on frequent interaction.

**Organizational and managerial aspects.** Recent work has pointed to how research on distance also has to include concerns for the organizational and managerial aspects of these setups. The diversity of organizational structures in our cases also clearly demonstrates how organizational situations—for example, outsourcing—directly affect the collaboration. For instance, in situations where the bread and butter of the organization is to develop IT systems, outsourcing the core business from Denmark to India might be interpreted by the Danish developers as a threat to their jobs, thus negatively affecting the conditions for developing common ground. This is different in situations of offshoring, where the success of the distributed team is a joint success for the company; thus, distributed team members are more willing to create the common ground related to the domain knowledge critical for success. In organizational setups where the cooperative engagement was characterized by strategic partnerships—meaning that if the outsourcing partner succeeds in fast delivery of quality systems, the client company would benefit financially, and vice versa—the conditions for creating strong foundations for distributed collaboration were best.

**Distance work in the future.** The future of distributed collaboration continues to evolve, initiating more complex organizational setups combining outsourcing, offshoring, and strategic partnerships working across multiple locations, partners, and time zones. People will continually be engaged in closely coupled work supported by collaborative technologies. There is increasing interest in applying software development methods that support such collaborative setups, for example, Scrum, which also pushes technologies to support closely coupled

work arrangements. However, one question remains: What does it mean for working conditions when we introduce approaches invented for the collocated setting into the global setting? What happens when Scrum goes global?

Our empirical data provides some answers for what might happen. In the INIT case, interviewing software developers working out of India in their third year revealed some interesting findings. The introduction of global Scrum combined with a multi-vendor outsourcing setup made the software developers' workdays increasingly stressful. The clients used the organizational structure and competition between vendors to push for shorter sprints and deadlines. In addition, all software developers had daily performance evaluations using the Scrum stand-up meetings as the tool for measurement, rather than empowerment. Also, the time-zone discontinuity of the project (India-U.S.) combined with the daily stand-up meetings meant the workday lasted until 10 p.m. in their time zone, five days a week (including Friday). The software developers expressed to us that they preferred Waterfall over Scrum; it is a very different answer from software developers engaged in collocated Scrum processes. Global Scrum carries the risk of people being perceived as assets rather than as human beings by creating stressful work environments with continuous work cycles, constantly pushing for more results faster, and forcing software developers to work out of sync with their own time zone, leaving little time for personal life and family.

Future research on distance collaboration must take the organizational aspect into account. This includes taking a stand for the politics of the workplace, where we should do our best to mitigate the risks of practices such as global Scrum and make sure that future workplaces do not lose the humanity, the self-organization, and the empowerments of the employee by reducing software development to a small part in a larger algorithmic machine, where the collaborative work resembles an assembly line in a factory. This, at least, is my hope for the future of distant collaboration.

## ACKNOWLEDGMENTS


The Next Generation of Tools and Processes for Global Software Development (NexGSD.org) research project is funded by the Danish Strategic Research Council and includes Pernille Bjørn (University of Copenhagen DIKU), Jakob Bardram (Danish Technical University DTU), Anne-Marie Søderberg (Copenhagen Business School CBS), Krishna (Indian Institute of Management, Bangalore IIM-B), and Ali Barbar (IT University of Copenhagen ITU). Partner companies include NNIT, KMD, L&T InfoTech, TCS, Teo, HCL, and Optivation. Ph.D. students and post-docs who contributed to the empirical work presented here include Rasmus Eskild Jensen, Stina Matthiesen, Morten Esbensen, Steven Jeuris, and Lars Rune Christensen. The work presented in this article is based on [2].

---

## ENDNOTES

1. Olson, G.M. and Olson, J.S. Distance Matters. *Human-Computer Interaction* 15, (2000), 139–178.
2. Bjørn, P., Esbensen, M., Jensen, R.E., and Matthiesen, S. Does distance still matter? Revisiting the CSCW fundamentals on distributed collaboration. *ACM Trans. on Computer Human Interaction (ToCHI)* 21, 5 (2014), 1–27.
3. Jensen, R.E. and Bjørn, P. Divergence and convergence in global software development: Cultural complexities as societal structures. *Proc. of COOP 2012*.
4. Søderberg, A-M, Krishna, S., and Bjørn, P. Global software development: Commitment, trust and cultural sensitivity in strategic partnerships. *Journal of International Management* 19, 4 (2013), 347–361.
5. Matthiesen, S., Bjørn, P., and Petersen, L.M. 'Figure out how to code with the hands of others': Recognizing cultural blind spots in global software development. *Proc. of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, New York, 2014, 1107–1119.
6. Esbensen, M. and Bjørn, P. Routine and standardization in global software development. *Proc. of GROUP 2014*. ACM, New York, 2014.

---

 **Pernille Bjørn** is a professor in CSCW in the computer science department at the University of Copenhagen, Denmark (DIKU). She explores collaborative practices with the aim of designing collaborative technologies and has recently been engaged with practice-based CSCW research, exploring how to comprehend and conceptualize transnational collaboration, in particular the politics behind conditions for work.  
→ [pernille.bjorn@di.ku.dk](mailto:pernille.bjorn@di.ku.dk)