Managing Distributed Software Projects

Persson, John Stouby

Publication date:
2010

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Persson, J. S. (2010). *Managing Distributed Software Projects*. Department of Computer Science, Aalborg University.

Ph.D. Thesis

Managing Distributed Software Projects

By

John Stouby Persson

Abstract

Increasingly, software projects are becoming geographically distributed, with limited face-to-face interaction between participants. These projects face particular challenges that need careful managerial attention. This PhD study reports on how we can understand and support the management of distributed software projects, based on a literature study and a case study. The main emphasis of the literature study was on how to support the management of distributed software projects, but also contributed to an understanding of these projects. The main emphasis of the case study was on how to understand the management of distributed software projects, but also contributed to supporting the management of these projects.

The literature study integrates what we know about risks and risk-resolution techniques, into a framework for managing risks in distributed contexts. This framework was developed iteratively, using evaluations by practitioners. Subsequently, the framework served as the foundation for the design of a risk management process, compliant with Capability Maturity Model Integration's (CMMI) generic approach to risk management (2006).

The case study investigates the managerial challenges of control and coordination in a successful, distributed software project between a Russian and a Danish company. The case study's control aspects were investigated, drawing on Kirsch's (2004) elements of control framework, to analyze how control is enacted in the project. This analysis showed that informal measurement and evaluation controls were used even though the team was short-lived and rarely met face-to-face; in addition, informal roles and relationships, such as clan-like control, were also used. The investigation suggests that management in successful distributed software projects can be highly reliant on both formal and informal controls and in both a project context and mediated communications. The case study's coordination aspects were investigated, drawing on the collective mind concept, developed by Weick and Roberts (1993), to analyze the patterns of mediated interactions in multimodal communication. This analysis showed that multimodal communication can facilitate collective minding in distributed software projects and can positively impact performance. In providing an approach for investigating the impact of multimodal communication practices on virtual team performance, we can further understand and support coordination in distributed software projects.

Keywords: Distributed Software Projects, Project Management, Risk Management, Control, Coordination

Resume

Softwareprojekter bliver oftere geografisk distribueret med begrænset ansigt til ansigt interaktion mellem deltagerne. Disse projekter står overfor specielle udfordringer, der kræver særlig ledelsesopmærksomhed. Dette PhD-studie rapporterer om, hvordan vi kan forstå og understøtte ledelsen af distribuerede softwareprojekter. Studiet er baseret på et litteraturstudie og et casestudie. Litteraturstudiets primære fokus var hvordan ledelse af distribuerede software projekter kan understøttes. Yderligere har studiet også bidraget til en forståelse af disse projekter. Casestudiets primære fokus var på hvordan ledelse af distribuerede softwareprojekter kan forstås, men har også bidraget til understøttelse af disse projekters ledelse.

Litteraturstudiet integrerer hvad vi ved om risici og risikoløsningsteknikker i et framework for risikoledelse i distribuerede kontekster. Dette framwork var udviklet iterativt med hjælp af praktiker evalueringer. Efterfølgende udgjorde frameworket fundamentet for designet af en risikoledelsesproces tilpasset *Capability Maturity Model Integration's* (CMMI) generiske risikoledelses tilgang (2006).

Casestudiet undersøger ledelsesudfordringerne forbundet med kontrol og koordinering i et succesfuldt distribueret softwareprojekt mellem en russisk og en dansk virksomhed. Casestudiets kontrolaspekter var undersøgt med Kirschs (2004) framework for kontrolelementer til at analysere, hvordan kontrol udøves i projektet. Denne analyse viste, at uformel målings- og evalueringskontrol blev anvendt, på trods af at teamet havde en kort levetid og sjældent mødtes ansigt til ansigt. Derudover blev der også anvendt uformelle roller og relationer såsom klanlignende kontrol. Undersøgelsen indikerer, at ledelse i succesfulde distribuerede softwareprojekter kan være dybt afhængig af både formel og uformel kontrol i både projektkonteksten og den medierede kommunikation. Casestudiets koordineringsaspekter var undersøgt med *collective mind* konceptet, udviklet af Weick og Roberts (1993), til at analysere medieret interaktions mønstre i multimodal kommunikation. Denne analyse viste, at multimodal kommunikation kan understøtte *collective minding* og have en præstationsfremmende effekt i distribuerede softwareprojekter. Med den foreslåede metode til at undersøge effekten af multimodal kommunikationspraksis på præstation i virtuelle teams kan vi yderligere forstå og støtte koordinering i distribuerede softwareprojekter.

Nøgleord: Distribuerede softwareprojekter, projektledelse, risikoledelse, kontrol, koordinering

Acknowledgements

I thank Jesper Boeg, Thomas S. Madsen and Flemming Steinson for co-authoring the first paper in my thesis "*Managing Risks in Distributed Software Projects: An Integrative Framework*"; Ivan Aaen for his thought provoking advising during my PhD study and co-authoring the third paper "*Enacting Control through Media and Context in Agile Distributed Software Development*"; and Lars Mathiassen for his inspirational and energetic advising along with co-authoring the four research papers in my thesis. Ivan Aaen provided face-to-face advice at Aalborg University while Lars Mathiassen at Georgia State University gave me first-hand experience in long distance collaboration with information and communication technology. I also thank the participants in the *Comapping* projects for allowing me to investigate their work practices, along with the companies who helped evaluate the risk management artifact. Finally, I thank my wonderful wife Anne for her invaluable support over the last three years.

John Stouby Persson

August 2009.

Table of Content

1 Introduction

This chapter describes the motivation for the PhD study's topic and research questions and it summarizes the thesis structure. The motivation for the study is grounded in the globalization of software businesses, resulting in an increased need for collaboration across geographical distance. This type of collaboration in software projects encompasses significant and persistent challenges. This PhD study therefore, examines how to understand and support the management of distributed software projects.

1.1 Motivation

We are confronted with globalization in both our professional and private lives. Globalization is a term describing the process of transforming local phenomena into global phenomena. The term is sometimes used to describe threats, or prosperity. An example of a threat could be that a person in a developing country will take your job. Prosperity could be seen as globalization, increasing accessibility of consumer goods previously only reserved for the privileged few. Globalization also raises moral issues about how we act as a global society. For example, concern has been raised that globalization causes exploitation of underdeveloped countries. Alternatively it can be argued that this engagement in underdeveloped countries will stimulate prosperity in these countries. Whether the optimistic or pessimistic view is adopted it is what we do in response to globalization that is important, particularly within the software industry.

Software professionals have played a significant role in relation to globalization. While globalization, in the past, was facilitated by developments in transportation, advances in software are now also key facilitators. Software has facilitated globalization by making the exchange of a wide range of services easy and fast across geographical distances. In fact, software has been so successful in facilitating such exchanges, that the service of developing software itself can be exchanged between distant places.

In general, the software industry appears to have adopted a positive view of globalization. This embracement has made relocation of white-collar software development jobs increasingly common. In the past, job relocation due to globalization primarily affected blue-collar jobs. However, in contrast to the blue-collar job relocations in the past, today, many of the white-collar jobs are only partly relocated. Partial job relocation leaves a new need for collaboration across geographical distance. Collaboration, facilitated by information and communication technology (ICT) between software professionals across geographical distance, is therefore, increasingly important. However it is unclear, how the many software professionals currently involved in this practice are, and possibly could be, responding to this change.

1.2 Background

The internet and associated technologies have made it possible to exchange data in real-time across the globe. This possibility has opened several new communication channels, such as instant messaging, net-meetings, and video conferences. These technologies have allowed team operations to move beyond co-located situations to teleconferences and on to virtual conferences (Bergiel et al. 2008). Such communication capabilities have made it increasingly feasible to have software projects with participants collaborating closely across geographical distance (Rutkowski et al. 2002; Sarker et al. 2005). This geographical distribution becomes increasingly frequent, with a massive transfer of development activities from the United States (US) and Europe to developing countries (Meyer 2006). In fact, information technology and product development are the functions, of which the highest cumulative percentage of US companies are initiating off-shoring (Manning et al. 2008).

The reason for distributing software development can be to improve time-to-market by round-the-clock development or to increase flexibility in capitalizing on merger and acquisition opportunities (Herbsleb & Moitra 2001). Other reasons include access to cheaper labor (Nicholson & Sahay 2004), increasing knowledge of customers and local conditions by market proximity, or capitalizing on the global talent pool (Conchúir et al. 2009; Damian & Moitra 2006; Herbsleb & Moitra 2001). In fact, shortage of high skilled science and engineering talent and, more generally, needs for access to qualified personnel are important explanatory factors for off-shoring innovation decisions (Lewin et al. 2008; Manning et al. 2008). In line with these trends, distributed software development is now no longer only an option for most enterprises; rather, it is a business necessity (Damian et al. 2008).

The developments in ICT provide new business and collaboration opportunities for software projects, but they come at a cost. ICT-enabled environments "*magnify the differences between good and bad projects, organizations, teams, and leaders. The nature of such projects is that there is little tolerance for ineffective leadership*" (Cascio & Shurygailo 2003). The task of developing software is a very coordination- and communication-intensive activity (Kraut & Streeter 1995). With coordination and communication proving difficult in distributed settings (Powell et al. 2004), project management becomes even more challenging. Indeed, distributed software development has unique nuances, complexities, and challenges. These characteristics range from technical, economic, organizational, and cultural issues, to those arising from different time zones, languages, and geographical locations (Damian & Moitra 2006). Making distributed project teams work effectively, and delivering quality systems on time and within budget, is thus a significant industry-wide challenge (Espinosa et al. 2006). In response to these challenges, researchers and practitioners are continuously developing vast amounts of guidelines, tools, and methodologies. While some resulting efforts can provide great benefits, these efforts often induce or redirect attention to new difficulties. In line with the software engineering silver bullet conundrum, new solutions induce new challenges (Berry 2008). Consequently, now more than ever, there is a need for knowledge on the increasingly widespread yet immensely complex endeavor of managing distributed software projects.

1.3 Research Questions

The growth in globally-distributed software development has increased the need for understanding appropriate engineering and management approaches (Damian & Moitra 2006). Consequently, this PhD study investigates how we can *understand* and *support* the management of distributed software projects.

The pursuit of knowledge in this study is guided by the research knowledge and activity framework proposed for collaborative practice research (Mathiassen 2002). This framework states that *understanding* should be based on interpretations of practice. This includes collecting data about practice and interpreting these using different concepts and frameworks. The outcome is an insight into practice, including concepts and frameworks that can be used to reflect upon practice (Mathiassen 2002). These activities are guided by the first research question (RQ):

RQ1: How can we understand management of distributed software projects?

The framework also states that knowledge of how to *support* practice should be based on making normative propositions or designing artifacts. The concern here is to create knowledge that can be used to plan, guide, or improve practice. The outcome can be in the form of some artifact[1]. The artifact should be developed and tested in relation to particular systems development disciplines (Mathiassen 2002). These activities are guided by the second research question:

RQ2: How can we support management of distributed software projects?

The two research questions are mutually dependent in the sense that the answer to either research question can influence the other. For instance, the support of advanced ICT made distributed teamwork significantly more feasible (Xue et al. 2004). However, this support also necessitated a new understanding of the phenomenon. This PhD study therefore, includes two different research efforts addressing the two research questions, however, with particular emphasis on either RQ1 or RQ2. The first research effort had an initial emphasis on RQ2, while also contributing to RQ1. The second research effort had an initial emphasis on RQ1 while also contributing to RQ2. Furthermore, the two research questions include the pronoun "*we*", which not only refers to researchers interpreting management practices in distributed software projects, but also practitioners engaged in it. Practitioner understanding and support needs to be taken into account in order to provide comprehensive answers for the two research questions (Mathiassen 2002). This PhD study therefore offers both a practice-based and a research-based investigation of the two research questions.

1.4 Structure of the Thesis

The thesis consists of four individually published papers and this summary. An elaborate research presentation should account for the 1) area of concern, 2) research questions, 3) frameworks, 4) methodology, and 5) contribution (Checkland & Holwell 1998). This PhD study's *area of concern* is the management of distributed software projects, presented in Chapter 2. The *frameworks* for investigating management of distributed software projects are risk management, control, and coordination, and are presented in Chapter 3. The *methodology* used for the investigation used a pluralistic research approach including a literature study and a case study, presented in Chapter 4. This PhD study's *contribution* contains four research papers, each addressing the two *research questions*, which is presented in Chapter 5. Finally, contributions, implications, limitations, and future research are discussed in Chapter 6.

[1] In this PhD study, the term artifact is also used to cover information systems.

2 Distributed Software Projects

This chapter presents existing research on the management of distributed software projects. In developing a common frame of reference concerning management of distributed software projects, the conceptual foundations in previous research is analyzed. Additionally, practice- and research-related challenges for managing distributed software projects are presented. Addressing these challenges is the primary area of concern for this PhD study.

2.1 Conceptual Foundations

The practice of geographically distributed collaborators in projects or organizations has been described by many different conceptualizations. While each conceptualization suggests different focus areas, the underlying observations of distributed collaborators are often similar. Four common conceptualizations are:

- Virtual teams (Gibson & Gibbs 2006; Orsak & Etter 1996; Powell et al. 2004)
- Offshore outsourcing (Kaiser & Hawk 2004; Pfannenstein & Tsai 2004; Smith et al. 1996)
- Virtual organizations (Bleecker 1994; Markus et al. 2000; Mowshowitz 1997)
- Global software development (Damian & Moitra 2006; Gao et al. 1999; Herbsleb & Moitra 2001)

Indicated by several literature studies in different research fields, virtual teams is a widespread and frequently used conceptualization (Curseu et al. 2008; Gillam & Oppenheim 2006; Hertel et al. 2005; Kirkman & Mathieu 2005; Martins et al. 2004; Powell et al. 2004; Schiller & Mandviwalla 2007). However, as the literature on virtual teams has grown, many different definitions have appeared. The foundation for the majority of definitions is the notion that virtual teams are functioning teams that rely on technology-mediated communication while crossing several different boundaries (Martins et al. 2004). Commonly-noted boundaries are geographic, time, and organizational dispersion, while additional characteristics are electronic dependence, structural dynamism, and national diversity (Gibson & Gibbs 2006; Martins et al. 2004; Powell et al. 2004). Furthermore, the term "team" suggests that such collaborating individuals are groups that display high levels of interdependency and integration (Powell et al. 2004). Virtual teams are often assembled from different organizations via outsourcing, or through joint ventures crossing organizational boundaries (Martins et al. 2004; Townsend et al. 1998; Zigurs 2003). When virtual teams are assembled from different organizations, it is also often conceptualized as offshore outsourcing.

The offshore outsourcing conceptualization suggests a particular emphasis on cross-organizational transactions, by the term "outsourcing". The term "outsourcing" reflects the use of external agents to perform one or more organizational activities (e.g., purchasing of a good or service) (Dibbern et al. 2004). This can apply to everything from the use of contract programmers to third-party facilities management. Additionally, the term "offshore" emphasizes a crossing of national borders. The term "offshore" furthermore reflects outsourcing to countries other than those that have traditionally dominated the software development industry (Smith et al. 1996). Offshore outsourcing arrangements can include a virtual team setting, but it is only one of many approaches (Dibbern et al. 2004). Some offshore outsourcing cases pursue high levels of interdependency and integration, which is compliant with virtual teams, while others go in opposite directions (Dibbern et al. 2004; Kaiser & Hawk 2004).

Cross-company organizations and the emergence of organizations with extensive reliance on ICT have also been conceptualized as virtual organizations. Virtual organizations are defined as flexible networks of independent, globally-distributed entities (individuals or institutions) that share knowledge and resources and work toward a common goal (Ripeanu et al. 2008). Virtual organizations are thus, conceptually different from offshore outsourcing, which is limited to cross-company transactions. Virtual organizations are also different from virtual teams, which are limited to groups with distributed members that display high levels of interdependency and integration. The open-source software movement has, in particular, provided inspiration for the development of the virtual organization conceptualization (Markus et al. 2000).

Software development by distributed collaborators has also been conceptualized as “global software development”. This conceptualization is based on the observation that software development is increasingly a multisite, multicultural, globally-distributed undertaking (Herbsleb & Moitra 2001). Conceptually, global software development can therefore, be compliant with all of the three concepts mentioned above. For instance, the emphasis of extensive technology-supported teamwork and communication in global software development research (Ebert & De Neve 2001; Wolf et al. 2008) is similar to much of the virtual team research. Other global software development research focuses on going beyond communication technologies by reducing intensive collaboration (Carmel & Agarwal 2001). This approach is also suggested in the offshore outsourcing and virtual organization research. However, in general, “global software development” does not appear as established or clearly defined a concept as “virtual teams”, “offshore outsourcing”, and “virtual organizations”.

The conceptualization used in this PhD study is the “distributed software project”. A distributed software project can be understood as a virtual team, an offshore outsourcing arrangement, a virtual organization, and a global software development effort. Each of the four concepts therefore, encompasses relevant research when investigating the management of distributed software projects. Consequently, all four lines of research are included in the body of knowledge for this thesis. However, the research topic emphasizes “software projects”, which often rely on the use of teams when developing software. The PhD study therefore, has a particular focus on virtual teams and global software development research.

2.2 Practical Challenges

Virtual teamwork characteristics, such as geographic dispersion, electronic dependence, structural dynamism, and national diversity often hinder innovation (Gibson & Gibbs 2006). This observation applies specifically to virtual teams developing software and implies numerous management challenges (Iacovou & Nakatsu 2008; Sakthivel 2007).

The challenges of managing distributed software projects arise because the project task is divided and distributed across several sites. The task division and distribution can make it difficult for project participants to understand the task, its purpose (Kirkman et al. 2004; Sakthivel 2005), and their own contribution to the overall task (Ebert & De Neve 2001; Farshchian 2001; Herbsleb & Mockus 2003). Consideration of interfaces, subsystem influence, and workload is therefore, critical when segmenting the task in distributed software projects. An increased coupling between task segments can increase the need for inter-site communication, coordination, and integration and may thereby lead to an increased number of failures (Carmel & Agarwal 2001; Herbsleb & Grinter 1999; Sarker & Sahay 2004). Also, when software developers from different parts of the world collaborate, tool compatibility is a common challenge. The reason is that each site is likely to prefer different programming languages, support tools, operating systems, and development tools (Dubé & Paré 2001; Kayworth & Leidner 2000; Sarker & Sahay 2004). Selection of appropriate ICT is therefore, significant for project success (Kayworth & Leidner 2000; Majchrzak et al. 2005).

Communication is a critical challenge in software project management (Gopal et al. 2002). In a distributed project, communication is even more difficult when participants are distributed across time and space. Social interaction is often impeded by low information exchange (Alge et al. 2003), absence of informal communication (Herbsleb & Grinter 1999; Herbsleb & Moitra 2001), and lack of face-to-face interaction (Dubé & Paré 2001; Pauleen 2003). This can negatively impact trust (Kayworth & Leidner 2001; Sarker & Sahay 2004), decision quality (Herbsleb & Moitra 2001; Kayworth & Leidner 2001), creativity (Kayworth & Leidner 2001; Sarker & Sahay 2004), and management (DeRosa et al. 2004). Furthermore, communication limitations can reduce participants' project overview and lead to errors and misunderstandings (Herbsleb & Moitra 2001). When the interaction media limits verbal and nonverbal cues, it is not possible to apply traditional conference management to virtual meetings (Sarker & Sahay 2004; Warkentin et al. 1997). Communication technology properties or use may therefore, cause problems. This can include problems, such as jumbled sequences of messages; mix-ups between past, present, and future messages (Massey et al. 2003; Sarker & Sahay 2004); and loss of contextual information sharing (Kayworth & Leidner 2001; Warkentin & Beranek 1999).

Distributed project participants may not share language, traditions, or organizational culture. Language barriers are typically present in cross national projects when sites and participants do not share a common native language (Krishna et al. 2004; Pauleen & Yoong 2001) or norms of communication (Kayworth & Leidner 2001; Townsend et al. 1998). These barriers can result in misinterpretations and un-conveyed information (Dubé & Paré 2001; Pauleen & Yoong 2001; Sarker & Sahay 2004). Differences in work culture may also lead to difficulties in a distributed software project. These difficulties can appear when sites are different in terms of team behavior or organizational culture (Carmel & Agarwal 2001; Chudoba et al. 2005; Dubé & Paré 2001). The difficulties can be based on differences between sites, in balancing collectivism and individualism, perception of authority and hierarchy (Herbsleb & Moitra 2001; Krishna et al. 2004), and planning and punctuality (Herbsleb & Moitra 2001). This may lead to decreased conflict-handling capabilities, lower efficiency, or even paralyze a distributed software project (Dubé & Paré 2001; Mortensen & Hinds 2001; Paul et al. 2004).

In general, when projects are distributed across time, space, and culture, it is difficult to obtain the same level of group cohesion expected in collocated teams (Sakthivel 2005). Project participants are less likely to commit to the project organization and its task, when cultural differences and lack of face-to-face interaction makes it difficult to establish a clear project identity (Bell & Kozlowski 2002; Furst et al. 1999; Mortensen & Hinds 2001). This weakens group synergy (Bell & Kozlowski 2002; Furst et al. 1999), increases the risk of conflicts (Mortensen & Hinds 2001), and may lower efficiency in the initial project phase (Tan et al. 2000). Distributed software projects are often characterized by horizontal organizational structures (Breu & Hemingway 2004). Flexibility concerning roles and assignments is therefore an important quality (Townsend et al. 1998). Processes, in terms of traditions, development methods, and emphasis on user involvement will often differentiate between sites, possibly resulting in incompatibility and goal conflicts (Battin et al. 2001; Evaristo et al. 2004; Sakthivel 2005).

The many challenges inherent in distributed software projects, which have already been mentioned, have motivated a large amount of research. Yet, many of the challenges are very persistent, and many research challenges in relation to distributed software projects still need to be addressed. This PhD study will investigate the mentioned practical challenges in distributed software projects using two research efforts. The first research effort is a categorization of known challenges and resolutions related to distributed software projects, along with guidelines for how to apply these categorizations in practice. The second research effort is an in-depth investigation into practice in a distributed software project with particular emphasis on a subset of the mentioned practical challenges. The research challenges addressed by these two research efforts are specified in the following.

2.3 Research Challenges

Much of the early work done in the area of project management does not address the problems existing when developers are located in geographically distributed places (Evaristo et al. 2004). However, recently, a wide range of alleviating initiatives has been suggested in response to the many challenges in managing distributed software projects. Obtaining an overview of the many challenges and their possible resolutions is, however, difficult. Furthermore, while many valuable contributions are available, no comprehensive approach to effectively manage the challenges in distributed software projects has been developed (Powell et al. 2004). Consequently, there is a need to understand and support the integration of existing knowledge to develop comprehensive approaches to manage effectively the challenges in distributed software projects. This research challenge is addressed in this PhD study's first research effort focusing on risk management documented in research papers 1 and 2.

The agile development methodology is a comprehensive approach, recently adopted to address distribution challenges in software projects (Holmström et al. 2006; Pries-Heje et al. 2005). The agile methodology is appropriate despite the underlying principle to convey information to, and within, a development team through face-to-face conversations. The introduction of agile development methodology in distributed software projects also implies attention to several concerns. Agile distributed software projects require particular attention to controlling the process and quality across sites (Ramesh et al. 2006). The agile characteristics of lightweight processes, ongoing negotiation, and reliance on skilled people therefore, induce new challenges related to balancing people- versus process-oriented control (Ramesh et al. 2006). In light of this observation, there is a need for increased research attention to understand and support control in the context of agile distributed software projects. This research challenge is addressed in the PhD study's second research effort focusing on control enactment through mediated communication and project context, documented in research paper 3.

With the rise of agile methodology in distributed software projects (Agerfalk & Fitzgerald 2006; Pries-Heje et al. 2005) comes increased attention to teamwork. Teamwork is characterized by high levels of interdependency and integration (Powell et al. 2004), which suggests a pronounced requirement for efficient support from ICT to coordinate activities. Furthermore, it is well documented that distributed projects face significant coordination challenges (Espinosa et al. 2007; Kanawattanachai & Yoo 2007; Kommeren & Parviainen 2007). While we have some knowledge of coordination with ICT, there is a need to know more on how ICT can successfully mediate coordination and help address these challenges. This is especially the case for agile methodology as it emphasizes high interdependency and integration, which has proved very difficult to maintain in distributed settings, where there is much reliance on ICT (Gibson & Gibbs 2006). On the basis of this difficulty, there is a need for increased research attention to understand and support successful coordination with ICT in distributed software projects. These research challenges are addressed in the PhD study's second research effort focusing on coordination through collective minding with multimodal communication, documented in research paper 4.

In summary, this PhD study investigates how we can understand (RQ1) and support (RQ2) the management of distributed software projects. The research questions are pursued by two studies. Each study emphasizes one of the research questions while also contributing to the other. The first study particularly focuses on RQ2 in the development of a comprehensive approach for managing the challenges inherent to distributed software projects. The aim of this comprehensive approach is to integrate systematically what we know of the challenges in managing distributed software projects and the resolutions to these challenges. This integration of challenges and resolutions also contributes to our understanding of managing distributed software projects (RQ1). The second study particularly focuses on RQ1 in the investigation of agile principles' contribution to the fundamental project management challenges in control and coordination with ICT in distributed settings. Understanding how control and coordination can be done with success in distributed software projects, underlie propositions for how this can be supported (RQ2).

3 Theoretical Framing

This chapter presents the PhD study's theoretical frameworks: risk management, control, and coordination. Each of the three frameworks features a fundamental project management challenge. This chapter explains what the three theoretical frameworks are and how previous research has investigated the related topics. Furthermore, it explains how the frameworks, used in this PhD study, build on previous research.

3.1 Risk Management

The need to understand and support the management of distributed software projects by integrating existing knowledge and developing comprehensive approaches, spurred on the use of risk management. The use of risk management as theoretical framing is reasoned by its successful application in collocated software development (Iversen et al. 2004; Lyytinen et al. 1998). Yet, the traditional risk management approaches fail to address the unique challenges that distinguish distributed from collocated software projects. The need for increased attention to risk management in distributed settings is furthermore reflected in a significant amount of recent research (Ebert et al. 2008; Erickson & Evaristo 2006; Iacovou & Nakatsu 2008; Nakatsu & Iacovou 2009; Prikladnicki et al. 2006; Sakthivel 2007).

Risk management can help practitioners assess problematic aspects of a project by emphasizing potential causes of failure, linking potential threats to possible actions, and facilitating a shared project perception among participants (Lyytinen et al. 1996; Lyytinen et al. 1998). These characteristics of risk management can help in developing an overview of the many challenges in managing distributed software projects and their possible resolutions in the literature. A comprehensive risk management approach, providing an overview of existing knowledge can thus be developed by identifying the causes of failures, suggested actions, and their mutual link documented in the relevant literature.

A software risk is a cause of failure in software projects, which denotes an aspect of a development task, process, or environment (Lyytinen et al. 1998). Ignoring a software risk increases the likelihood of project failure. The degree of risk can be assessed either quantitatively or qualitatively (Barki et al. 1993; Boehm 1991). A quantitative risk assessment is the probability of unsatisfactory events multiplied by the loss associated with their outcome. A qualitative risk assessment is done by referring to the uncertainty surrounding the project and the magnitude of potential loss associated with project failure.

The actions for addressing risks and the link between risk and action is known as heuristics. Heuristics in risk management have different appreciations, depending on the approach. Iversen et al. (2004) identify four different ways in which approaches to software risk management address risk items, resolution actions, and heuristics. These approaches are the:

1. Risk list, which is a list of prioritized risk items (Barki et al. 1993; Moynihan 1996; Ropponen & Lyytinen 2000).
2. Risk-action list, which is a list of prioritized risk items with related resolution actions (Alter & Ginzberg 1978; Boehm 1991; Jones 1994; Ould 1999).
3. Risk-strategy model, which is a contingency model that relates aggregate risk items to aggregate resolution actions (Donaldson & Siegel 2001; Keil et al. 1998; McFarlan 1981).
4. Risk-strategy analysis, which is a stepwise process that links a detailed understanding of risks to an overall risk management strategy (Davis 1982; Mathiassen et al. 2000).

Each of the four approaches has several advantages and some disadvantages (Iversen et al. 2004). The disadvantages of the four are: 1) the risk list lacks risk resolution and strategic oversight; 2) the risk action-list lacks strategic oversight; 3) the risk-strategy model is difficult to build and modify; and 4) the risk-strategy analysis is difficult to use and build.

The disadvantages of each approach were considered and then a decision was made regarding which of the four approaches would be chosen to integrate existing knowledge on managing distributed software projects. The risk list approach's lack of risk resolution excludes important knowledge on managing distributed software projects. The risk-strategy model's modification difficulty does not take enough consideration of the high-paced developments in distributed software project practice and research. The risk-strategy analysis's limited usability is problematic as the use of the approach is likely to be even more difficult in distributed and cross-cultural settings. The risk-action list tradeoff in lack of strategic oversight is less important compared to the other qualities because its adoption focuses on risks related to distribution rather than on risks in general. Based on these considerations, the risk-action list was chosen as the most appropriate way of integrating existing knowledge on managing distributed software projects into a comprehensive approach. The development of a risk-action list in this PhD study was more specifically based on Boehm's (1991) *software risk framework*, which is a classical risk management implementation(Lyytinen et al. 1998). Another risk management implementation, popular among software practitioners, is found in the process standard CMMI for development (CMMI-Product-Team 2006). In light of CMMI's popularity, it was also taken into consideration in the development of a comprehensive approach for managing risks inherent to distributed software projects.

CMMI (2006) is a maturity model that organizations can use to assess and improve their processes. The model offers a comprehensive set of generic processes to support development of products and services. One of the generic processes focuses on risk management and helps identify and analyze potential problems before they occur so that risk-handling activities can be planned and invoked across a project's lifecycle. The aim of this process is to effectively anticipate and mitigate the risks that may have a critical impact on the project. CMMI and its risk management process have successfully penetrated the software industry, in particular, large global organizations and companies in the popular offshore outsourcing destination, India. CMMI (2006) compliance considerations are therefore important when developing a comprehensive approach for such distributed settings.

3.2 Control

Understanding and supporting the management of distributed software projects was also addressed by investigating the increasingly popular agile approaches (Agerfalk & Fitzgerald 2006; Holmström et al. 2006; Pries-Heje et al. 2005). The investigation was motivated by agile methodologies' increasing popularity combined with the apparent incompatibility between their characteristics and a distributed work environment. Particular attention was given to the incompatibility regarding the fundamental management issue of control (Ramesh et al. 2006).

Broadly speaking, control means any attempt to motivate individuals to behave in a manner consistent with organizational objectives (Kirsch 2004; Ouchi 1979). The control perspective has been advocated by Kirsch (1996; 1997; 2000; 2004) in software development management research, and is based on the work by Ouchi (1978; 1979; 1980) and Eisenhardt (1985). Control is either formal or informal. Formal control is viewed by organizational researchers as a performance evaluation strategy, where behaviors or outcomes are measured, evaluated, and rewarded (Eisenhardt 1985; Kirsch 1996). Informal control differs from formal control in that it is based on social and people strategies (Eisenhardt 1985; Kirsch 1996).

Formal and informal control is related to four elements of control (Kirsch 2004). These four elements are *measurement*, *evaluation*, *rewards and sanctions* (Eisenhardt 1985), and *roles and relationships* (Kirsch 2004), which can be formal or informal. Formal measurement implies that behaviors or outcomes are explicitly specified and measurable, while informal measurement is when norms, values, or behaviors are implicitly specified and measured. Formal evaluation is based on specific information regarding behavior and outcome, and assesses if the existing status leads to progress. Informal evaluation refers to norms and values achieved by socialization through dialog or discussions, which are assumed to lead to performance. Formal rewards and sanctions are based on achieving specific goals or adhering to pre-specified behavior, by rewards such as bonuses or sanctions, such as demotions. Informal rewards and sanctions enforce behavior consistent with group norms and values by rewarding with, for example peer-recognition or sanctioning with social exclusion. Formal roles and relationships imply particular roles and usually a focus on dyadic relationships. Informal roles and relationships appear in groups of individuals dependent on each other and committed to group goals (Kirsch 2004).

Different elements or mechanisms of control are not applied separately (Ouchi 1979), but in portfolios to support management practice in specific contexts (Kirsch 1997). The creation, composition, and change of control mechanisms in software development are dependent on several influential factors. The process of constructing a portfolio of control mechanisms depends on task characteristics, role expectations and project-related knowledge, and skills (Kirsch 1997). Further influential factors may also be behavior observability, outcome measurability (Kirsch et al. 2002), cultural dimensions (Narayanaswamy & Henry 2005), the specific phase in the software project (Kirsch 2004), and even previously adopted control mechanisms (Cardinal et al. 2004).

While studies of control in distributed contexts have been conducted in relation to issues, such as trust (Piccoli & Ives 2003), culture (Narayanaswamy & Henry 2005), and effectiveness (Piccoli & Ives 2003), Kirsch (2004) calls for research to more closely examine the role of the global context on control choices and impacts. Further calls for research on control in distributed contexts more specifically question whether informal control mechanisms, in general can be used when teams are short-lived and rarely meet face-to-face (Powell et al. 2004). This concern is particular relevant when agile methodologies are adopted, as they mainly rely on face-to-face communication. In addition, Harris et al. (2006) argue that informal roles and relationships, such as clan-like control inherent in agile development, are likely to be more difficult to practice in a distributed setting. It is further argued that clan control can be increasingly difficult when interaction is not face-to-face but mediated by technology and when participants come from different organizations (Harris et al. 2006). Ramesh et al. (2006) supports the idea that agile characteristics, combined with distributed settings, induce new control challenges. These challenges are related to balancing people versus process-oriented control and fixed, versus evolving, quality requirements. Despite all these apparent challenges, agile development has been carried out with success in distributed settings (Armour 2007; Farmer 2004; Paasivaara et al. 2008; Sutherland et al. 2007). However, from a control perspective, there is a need to understand how this is feasible.

3.3 Coordination

The increasing popularity of agile methodology in distributed software projects (Agerfalk & Fitzgerald 2006; Pries-Heje et al. 2005), requires increased attention to teamwork. Teamwork is characterized by high levels of interdependency and integration, which imply an extensive need for coordination. Coordination is defined as managing dependencies between activities (Malone & Crowston 1994). Successful coordination is characterized by integration and harmonious adjustment of individual activities towards the accomplishment of a larger goal (Singh 1992) or simply by working together effectively (Malone & Crowston 1991).

Distributed software projects do however, even without the introduction of agile methodologies, face significant coordination challenges related to the characteristics often associated with distribution (Gibson & Gibbs 2006; Powell et al. 2004); these characteristics, include:

1) Geographic dispersion, which may cause coordination difficulties related to time zone differences (Massey et al. 2003; Montoya-Weiss et al. 2001), locally-situated knowledge (Sole & Edmondson 2002), and lack of presence awareness (Espinosa et al. 2007).
2) National diversity, which can cause coordination difficulties related to different communication routines (Maznevski & Chudoba 2000; Robey et al. 2000), linguistic differences (Kayworth & Leidner 2000), and weak interpersonal relationships (Kraut et al. 1998).
3) Structural dynamism, which may cause coordination difficulties related to problematic task coupling (Carmel & Agarwal 2001; Ramesh & Dennis 2002; Sakthivel 2005), low task awareness (Espinosa et al. 2007), and problematic inter-functional conflict resolution (Robey et al. 2000).
4) Electronic dependence, which can cause coordination difficulties related to limitations of informal ad hoc communications (Herbsleb & Grinter 1999) and organizational identification (Wiesenfeld et al. 1998).

Electronic dependence, in the form of ICT, is often the only coordination space available for distributed software projects and is therefore crucial to their success. ICT is both a challenge and a frequently-proposed solution to coordination difficulties in distributed settings. Researchers and practitioners have provided many ICT solutions to help coordination challenges in distributed software projects. However, in-depth knowledge of how ICT help address coordination challenges in real distributed software projects is still needed.

The need for efficient coordination is a particularly important challenge in software projects. Software projects may include systems and tasks so complex, that it would be impossible for any single team member to hold all the knowledge required to succeed. In such cases, professional knowledge is specialized and distributed requiring coordination based on the knowledge of several team members (Cannon-Bowers & Salas 2001). The ability to effectively coordinate and collaborate in complex organizations has been conceptualized as a socio-cognitive phenomenon through the notion of collective mind (Weick & Roberts 1993). The major claim of collective mind theory is that individuals facilitate group performance by developing shared understandings of a team’s tasks and of one another (Crowston & Kammerer 1998). In a collective mind, each act is heedfully interrelated with acts of other actors. Actors behave heedfully when they are careful, critical, consistent, purposeful, attentive, studious, vigilant, conscientious, and pertinacious. In a collective mind, each actor heedfully interrelates by subordinating, representing, and contributing to the social system, as they continuously coordinate activities (Weick & Roberts 1993).

The collective mind theory was developed by Weick and Roberts (1993) based on an investigation of the complex coordination of military aircraft carriers. The obvious difference between an aircraft carrier and an agile distributed software project may raise questions regarding the appropriateness of the collective mind framework. However, according to Weick and Roberts (1993), organic systems typically have more fully-developed minds than mechanistic systems because of their capacity to reconfigure themselves into dynamically-shifting structures. Agile software projects can, in the light of their dynamic characteristics be defined as organic systems.

There are also examples of underdeveloped collective minds, which include groupthink (Galanter 1989), the *Challenger* disaster (Starbuck & Milliken 1988), and ethnocentric research groups (Weick 1983). These examples share subordination with a social system that is envisaged carelessly (Weick & Roberts 1993). The examples furthermore underline the critical importance of ICT in distributed teams, since ICT often exclusively provides the basis for the envisioning of a social system.

The collective mind perspective has provided valuable insights into coordination in software requirements development (Crowston & Kammerer 1998), board meetings (Cooren 2004), and virtual team performance (Yoo & Kanawattanachai 2001). However, Yoo and Kanawattanachai (2001) call for field studies of the collective mind in distributed teams that focus on micro-level content analysis of mediated communication. Such analyses have proven valuable in Cooren's (2004) conversation analyses of board meetings. He proposes to focus on the process of collective minding, emphasizing that the collective mind can be found at multiple stages of development in an organization. Cooren's (2004) study provides valuable insights into how to investigate collective minding in organizational communication. Most importantly, he suggests that evidence of collective minding has to be found in interactional patterns and not only in the perception of individual actors.

4 Research Approach

This chapter presents the methodology used for this PhD study, which is a pluralistic research approach including a literature study and a case study. With the research questions, area of concern, and theoretical frameworks clarified, the choice of research approach can be elaborated. There is a need to understand and support distributed software projects by developing comprehensive approaches to effectively manage their inherent challenges (Evaristo et al. 2004; Powell et al. 2004). A comprehensive approach can be developed by integrating knowledge from relevant literature with a risk management framework. This chapter will present how such a literature study was carried out. Previous research also suggests increased attention to understanding and supporting control and coordination in distributed software projects (Fiore et al. 2003; Kirsch 2004; Powell et al. 2004; Yoo & Kanawattanachai 2001). These challenges can be understood and supported by investigating practices in successful distributed software projects. Therefore, this chapter will also present how such investigation of practice was carried out as part of this study.

4.1 Research Design

This PhD study investigates how we can understand (RQ1) and how we can support (RQ2) the management of distributed software projects. In accordance with the research questions, the overall goal of this PhD study is therefore to gain knowledge related to *understanding* and *support*. The concepts *understand* and *support* are adopted from the research knowledge and activity framework proposed for collaborative practice research (Mathiassen 2002).

RQ1 pursues *understanding*, which is developed by engaging in interpretations of practice (Mathiassen 2002). This can be done by collecting data about practice and interpreting it using different concepts and frameworks. According to the related research method framework by Braa and Vidgen (1999), a case study is the most suitable approach for acquiring this type of knowledge. Alternatively, *understanding* practice can be achieved by synthesizing existing empirical studies. The resulting *understanding* can then take the form of insights into practice, concepts, and frameworks that can be used to reflect upon practice (Mathiassen 2002).

RQ2 pursues *support*, which is developed by the creation of normative propositions or designing artifacts, that can be used to plan, guide or improve practice (Mathiassen 2002). According to Braa and Vidgen (1999), a field experiment is the most suitable approach for acquiring this type of knowledge. Alternatively, *support* for practice can be developed by synthesizing existing or new empirical studies. The resulting *support* can take the form of an artifact or propositions that has been developed and tested in relation to particular systems development disciplines (Mathiassen 2002).

The alternative approaches available for each of the two types of knowledge, points in the direction of a plural methodology approach as advocated by Mingers (2001). Mingers argues that research results will be richer and more reliable if several research methods are used. Different research approaches have strengths and weakness depending on both the studied phenomenon and its surrounding conditions. A pluralistic approach is therefore needed to deal with complexity and richness when studying practice. There are however, different types of multimethod research designs (Mingers 2001):

- Sequential, where methods are employed in sequence with results from one feeding into the later one.
- Parallel, where methods are carried out in parallel with results feeding into each other.

- Dominant (imperialist), where one method or methodology is the main approach with contribution(s) from the other(s).
- Multi-methodology, which is a combination of methods, embodying different paradigms, developed specifically for the task.
- Multilevel, which is research conducted simultaneously at different levels of an organization and using different methods.

This PhD study adopted a parallel multimethod research design with two research approaches. Each of the two research approaches has a particular emphasis on RQ1 and RQ2.

The first research approach was a literature study, which had particular emphasis on RQ2, but also contributed to RQ1. A literature study was chosen because of its capacity to help create artifacts that can *support* practice. Furthermore, the literature study was followed by several field evaluations of the concepts and framework encompassed in the created artifact. This was done because Braa and Vidgen (1999) emphasize field experiments when investigating how to *support* practice. The choice of a literature study approach was also reasoned by its compliance with risk management as theoretical framing. A literature study can provide an overview of the many documented challenges and their possible resolutions in the literature relevant to risk management in distributed software projects. The failure causes, suggested actions, and their mutual link documented in the literature, can come together in a much-needed approach for managing risks related to the distribution of software projects. The literature study can thus address the lack of comprehensive approaches to effectively manage the challenges in distributed software projects (Powell et al. 2004). Conceptualization of failure causes suggested actions and their mutual link, finally contributing to *understanding* the management of distributed software projects (RQ1).

The second research approach was a case study, which focused on RQ1, but also contributed to RQ2. A case study was chosen because of its capacity to help develop an *understanding* of practice in distributed software projects. This help is in the guidelines for collecting data about practice and interpreting these using different concepts and frameworks. In fact, the case study approach is explicitly suggested by Braa and Vidgen (1999) for developing *understanding* of practice. The choice of a case study was also reasoned by its compliance with both control and coordination as theoretical framings for achieving *understanding* of practice. In relation to control, a case study can provide rich insights into control enactment in the context of an agile distributed software project. A case study's capacity for providing in-depth descriptions of a phenomenon in its context is thereby exploited. These descriptions can address the calls for research on control in distributed settings (Kirsch 2004) and empirical studies of agile software development (Dybå & Dingsøyr 2008). In addition, the descriptions can provide insights on the apparent incompatibility between the characteristics of agile methodologies and distributed software projects regarding control (Ramesh et al. 2006). In relation to coordination, a case study can similarly provide in-depth descriptions of this phenomenon in its context, especially concerning how ICT can successfully support coordination in teams characterized by high interdependency and integration. This research effort can thereby address the calls for field studies about collective mind in distributed teams focusing on micro-level content analysis of mediated communication (Yoo & Kanawattanachai 2001). The case study can furthermore help to understand and support the known difficulties in coordinating distributed software projects (Espinosa et al. 2007; Herbsleb & Grinter 1999; Sakthivel 2005). Propositions for control and coordination in distributed settings contribute to *supporting* management of distributed software projects (RQ2).

The reason for using a multimethod research design was motivated by the opportunity for richer and more reliable results through data triangulation and method pluralism (Mingers 2001). Furthermore, the research design explicitly addresses the pronoun "*we*" in the research questions, by investigating both research- and practice-based data sources. This is done by combining a practice-based investigation in the form of a case study, with a research-based investigation in the form of a literature study. In summary, this research design provide distinct emphasis on the two forms of the respectively resulting *understanding* (RQ1) and *support* (RQ2) knowledge, suggested by Mathiassen (2002), see Table 1. The resulting knowledge concerning *understanding* can take the form of **insights** into practice, based on the case study. The *understanding* can also take the form of **concepts** that can be used to reflect upon practice, based on the literature study. The resulting knowledge concerning *support* can take the form of normative **propositions**, based on the case study. The *support* knowledge can also take the form of **artifacts**, which can be used to plan, guide, or improve practice, based on the literature study, see Table 1.

Research goal	Research approach: Literature study	Research approach: Case study
Understand (RQ1)	**Concepts** A research-based risk and process focus on managing distributed software projects.	**Insights** A practice-based control and coordination focus on managing distributed software projects
Support (RQ2)	**Artifacts** A research-based framework and process contributing to the management of risks in distributed software projects	**Propositions** A practice-based control and collective minding conceptualization, contributing to the management of distributed software projects

Table 1 Two-by-two matrix for research approaches and goals

Finally, pluralism was also considered in the two individual research approaches to further provide richer and more reliable results (Mingers 2001). The literature study adopted a sequential multimethod research design by feeding the results of a literature study into field evaluations. The case study adopted a parallel multimethod research design concerning the collection of data by, as in Mingers (2001) illustration, observation and recording of computer usage together with interviewing. Yet, the thorough adherence to the guidelines for a case study proposed by Yin (2003), suggest that a dominant multimethod research design is a more appropriate labeling. Finally, the two pluralistic research designs followed Braa and Vidgen's (1999) directions for developing the types of knowledge sought by the two research questions.

4.2 Literature Study

The literature study had a primary focus on addressing RQ2 by developing an artifact for *supporting* management of distributed software projects. However, RQ1 was also addressed by the development of concepts, which can help *understand* management of distributed software projects. According to Webster and Watson (2002) the primary goal of a literature study is to achieve a complete result focused on concepts. Thus, the two most important tasks are to decide how to identify the relevant literature and how to conceptually structure the analysis (Mathiassen et al. 2007; Weill & Olson 1989). Therefore, the analysis should finally result in a strong synthesis and evaluation (Webster & Watson 2002).

4.2.1 Identification

The Web of Science article database was searched for relevant articles published in 1995 or later. Distributed software project research, prior to 1995, was considered to be of less interest, since it was only with the development of communication and collaboration technology during the 1990s that distributed development was made feasible for entire projects (Xue et al. 2004). The resulting set of articles was limited to include the 500 most relevant according to the Web of Science analysis tool (Thomson.Scientific 2005). Following Webster and Watson (2002), this set of articles was further restricted to include only those published in rated *Management Information Systems* and *Management* research journals. This was based on the assumption that many challenges in distributed software projects are similar to the ones encountered within other industries involved in distributed projects. The list of rated journals was a result of a thorough examination of studies of journals in the two areas of research: Management Information Systems (Katerattanakul et al. 2003; Lowry et al. 2004; Rainer & Miller 2005; Whitman et al. 1999) and Management (Franke et al. 1990; Gomezmejia & Balkin 1992; Johnson & Podsakoff 1994). The resulting articles were evaluated, based on a detailed examination of abstracts and those of little or peripheral interest, were excluded from the set. To ensure that key articles in this area of research were included in the final set, the cited references of all articles appearing more than once were evaluated, following the procedure described above. This approach was combined with structured critique to further steer the selection of (the 72) articles that were reviewed (Weill & Olson 1989).

4.2.2 Review Structure

The first part of the review was to identify the risks that pose the most threat to distributed projects. According to Boehm (1991), risk areas consist of a number of related risk factors, which together, possess a threat to the project's success. Risk areas thus represent categories of risk factors, where the joint assessment of risk factors indicates whether the risk area might become a problem for a project. Risk areas were systematically synthesized inspired by the categories of risk areas used in key articles, with an overall perspective on distributed software projects (Bell & Kozlowski 2002; Herbsleb & Moitra 2001; Powell et al. 2004; Sakthivel 2005; Townsend et al. 1998). Leavitt's (1964) model was used, as suggested by Lyytinen et al. (1998), to provide clear foci for a distinct set of risk areas. A complete list of risk factors identified in the literature was aggregated and categorized according to the proposed risk areas. Finally, questions and criteria were developed to provide precise definitions of each risk factor.

The second part of the review focused on identifying and categorizing resolution techniques that address risks through managerial intervention (Section IV). As no independent categorization of resolution techniques was identified in the reviewed distributed software project literature, inspiration was found in the software risk management literature. McFarlan (1981) presents a generic software risk management framework that has proven its worth time and again over the past 25 years. McFarlan uses four categories of resolution techniques centered on basic project management disciplines and with a particular focus on integration. As integration is a major challenge in managing distributed software projects, McFarlan's framework was adapted to help structure available resolution techniques.

McFarlan's (1981) categories are: *internal integration*, consisting of techniques to support coordination and communication, internally in the project group; *external integration*, consisting of techniques to support coordination and communication with external stakeholders; *formal planning*, consisting of techniques to support planning; and finally, *formal control*, consisting of techniques to ensure that formal planning stays on track and is continuously updated in relation to project practices. The literature on distributed software projects is less concerned with the challenges of internal and external integration. Instead, there is considerable focus on how communication and collaboration efforts can be supported by various forms of ICT. Also, social integration is generally considered a key challenge, because the presence of several cultures in distributed software projects creates an environment significantly different from collocated projects. Furthermore, recent research has pointed out that control is not only formal but also informal in distributed software projects (Choudhury & Sabherwal 2003; Kirsch 2004). In light of this context, McFarlan's (1981) concepts were adapted to the following resolution technique categories: *planning, control, social integration,* and *technical integration*. These categories helped distinguish all the risk resolution techniques identified in the literature.

4.2.3 Synthesis and Evaluation

The identified risks and resolution techniques were integrated into a framework with an outset in Boehm's (1991) risk-action approach. The risk management framework, based on the literature study, was developed iteratively. The practical usefulness of each version was evaluated and the findings fed into the next iteration. In total, four evaluations were conducted, with increasing focus on practical usage: *Evaluation I* focused on the initial conceptualization of risks and risk resolutions through a focus group; *Evaluation II* focused on paper-based risk assessment and risk management through a focus group and a workshop; *Evaluation III* focused on tool-based risk management through a workshop; and, *Evaluation IV* focused on full-scale application of the tool with multiple participants in a real-world setting. All four evaluations were documented through field notes, audio recordings, and work documents. The iterative development-evaluation process was terminated at this point, as Evaluation IV only led to minor changes and all participants found the artifact useful and easy to use. Furthermore, evaluation IV did provide an illustrative foundation for a risk management process based on the risk management framework. However, the development of this risk management process was predominantly guided by CMMI (CMMI-Product-Team 2006; Kulpa & Johnson 2003). The artifact, developed in this literature study, thereby encompasses 1) a risk management framework developed from state-of-the-art literature on distributed software projects and empirical evaluations, 2) a web-based tool instantiation of the framework, which facilitated further empirical evaluations of practical usefulness, and 3) a CMMI (2006) compliant process for using the risk management framework and web-based tool. The use of the "artifact" concept is inspired from collaborative practice research (Mathiassen 2002) and the design science research approach (Hevner et al. 2004; Winter 2008).

4.3 Case Study

The practice study had a primary focus on addressing RQ1 by developing insights into *understanding* management of distributed software projects. However, RQ2 was also addressed by the development of propositions, which can help *support* the management of distributed software projects. This investigation of practice was done as an explanatory single-case study for several reasons.

- Management of distributed software projects is a contemporary phenomenon that needs further investigation in real-life contexts (Smite et al. 2008). A case study is well suited for such an investigation, especially since the boundaries between the phenomenon and context are not clearly evident (Benbasat et al. 1987; Yin 2003).
- This PhD study pursues explanatory knowledge, which is reflected in the use of how-questions. In its capacity of investigating operational links, an explanatory case study is well suited for developing such knowledge (Yin 2003).
- This PhD study had access to a unique and interesting case, which included a rare combination of project distribution and management based on agile principles across two companies. A single case study is appropriate when the case is rare (Yin 2003).
- The classical management issues of control and coordination are challenged by new significant problems in distributed settings (Fiore et al. 2003; Powell et al. 2004). A single case study is suitable for testing the boundaries of well-formed theory (Benbasat et al. 1987).

The investigated case was a distributed software project relying on agile principles. The project combined teleconferencing with real-time collaborative modeling in their control and coordination efforts. The team's task was to finalize the development of the mindmapping tool they used to support collaborative modeling. It was thereby possible to investigate how an agile distributed software project, highly dedicated to multimodal communication, would control and coordinate activities. The theoretical frameworks for control and coordination were iteratively compared with empirical evidence to understand how a distributed software project managed to successfully control and coordinate its efforts. Access to multiple data sources, including audio and video recordings of mediated activities, provided rich opportunities for rigorous analyses, based on both quantitative and qualitative analyses (Mingers 2001; Sherif et al. 2006).

4.3.1 The Case

The investigated distributed software project was a joint venture between a small Danish software company, Area9, and a Russian research and development (R&D) outsourcing provider, Lanit-Tercom. The project was conceived by the Danish company Area9 and established in 2006 by two medical doctors and two computer scientists, all of whom previously had management positions in another software company also relying heavily on offshore developers. The two companies had equal ownership, but made different contributions to the *Comapping* project. Lanit-Tercom initially assigned two developers to the project while Area9 provided management, architectural, and design expertise by providing two full-time members of staff.

Area9 based their development practices on agile principles, and claimed that distributed projects should be managed as if the developers were located in Denmark. This management practice required that the developers were able to work in an agile environment and have excellent communication and collaboration skills. Their agile practices included continuous integration, parallel development and testing, incremental design, code reviews, and sparse documentation. In the coordination between sites the team relied mainly on e-conferences held with teleconferencing via Skype (www.skype.com) combined with real-time collaborative modeling via the shared mindmapping tool (www.comapping.com). The *Comapping* project thereby used the mindmapping tool to manage the development of the same mindmapping tool. E-conferences were held between the Area9 board member (representing management), the joint venture Chief Executive Officer (CEO) (representing marketing), and the Russian systems development manager (representing product development). The conference language was English and all e-conferences took place within normal working hours as the time-zone difference between the Danish and Russian sites was only two hours.

Contact was initiated with the *Comapping* project in early 2007 and e-conferences between the Danish and Russian sites were thereafter systematically observed. The project reached a major milestone when a Fortune 500-company invested in the system. As a result, customization to the new partner's requirements became the primary objective and the *Comapping* project staff was increased to eight full-time developers. Our case study ended in August 2007 when this milestone was reached and the project was reorganized.

4.3.2 Data Collection

Data about the *Comapping* project were collected from January 2007 to the end of August 2007. The data collected consisted of video and audio recordings of the e-conferences and interviews with the project participants. During the e-conferences, the author was present, offsite, as a passive observer, while audio recording conversations and video recording the real-time collaborative modeling that took place during the e-conferences. In all, 10 observations on the e-conferences were made from February 2007 to July 2007. Even though the mindmapping tool had not been officially released when data collection started, all its basic functionality was, in general, working seamlessly. The observations ended when the project reached the major milestone, which was when the Fortune 500-company invested in the system.

During the period when e-conference observations were made, a number of semi-structured interviews were conducted with the project participants. The interviews were conducted because it is widely recognized that studies of collaborative technology in distributed software projects must be well embedded within organizational and work group contexts (Majchrzak et al. 2000). The people interviewed were the manager (Denmark), who was a board member of the *Comapping* project and director of Technology & Innovation at Area9; the developer (Russia), who was director of R&D for the *Comapping* project; the second developer (Russia), who was a software developer in the *Comapping* project; the marketer (Denmark), who was CEO for the *Comapping* joint-venture; and the chairman (Denmark) of the joint-venture and CEO of Area9. The interviews were initiated with a face-to-face meeting with the manager. After this first interview, the other full-time project members were interviewed via Skype. Towards the end of the e-conference observations, a new series of interviews were conducted with the project participants and the chairman. In total, the e-conference observations were supplemented with 11 interviews.

4.3.3 Analyses

Data analyses were conducted with Atlas.ti V5.5 (Hwang 2008; Muhr 2008) allowing coding to be placed directly on the e-conference recordings. The direct coding was chosen because a traditional transcription of recordings would not form a feasible approach due to the multimodal nature of the data.

In the control analysis, all 14 hours of interview and observation recordings were loaded into the software and carefully listened through. Exchanges and statements pertaining to control were identified and coded as one of the three original elements of control (measurement, evaluation, rewards and sanctions); they were also coded with one of the identified roles or relationship types (Kirsch 2004). Furthermore, one or more descriptive codes, related to the specifics of the element of control, were added to each exchange or statement. Finally, specific control statements and exchange transcriptions were brought forth for additional qualitative analysis.

In the coordination analysis, a particular focus was on the activities in the e-conferences. This focus was carried out because coordination is defined as managing dependencies between activities (Malone & Crowston 1994). Two types of activities were coded in the e-conference recordings. The first type of coded activity was *manipulation acts* based on the actors' use of the mindmapping tool for real-time collaborative modeling. The second type of coded activity was *articulation acts* based on how the participants acted through spoken language in the e-conferences. This language-action perspective was based on Searle's (1969; 1979) typology of speech acts in cooperative work (Winograd 1987). The dependencies between these activities were investigated using the coding of *communication breakdowns*. A communication breakdown causes a disruption in work practices, shifting the actors' attention towards an appropriate recovery strategy, which compromises or challenges coordination between actors (Bjørn & Ngwenyama 2009; Ngwenyama 1998). Coordination is most clearly noticeable when it is lacking (Malone & Crowston 1994), therefore, communication breakdown coding was used to investigate the management of dependencies between the activities. In the coordination analysis, each of the three coding schemes also served as indicators on Weick and Roberts' (1993) collective mind conceptualization: 1) Manipulation acts were primary indicators of the actors' representations of the social system, 2) articulation acts were primary indicators of the actors' contributions to the social system, and 3) communication breakdowns were primary indicators of actors' subordination to the social system.

The coordination analysis of audio and video recordings of seven e-conferences was triangulated with the interview data. The interview recordings were revisited multiple times throughout the research process. This was done in order to triangulate insights developed from the e-conference analysis with the actors' own perceptions. Thus the analysis was related to the organizational and work group contexts, as emphasized by Majchrzak et al. (2000) when investigating technology adoption in distributed settings.

5 Research Summary

This chapter presents the PhD study's contribution, consisting of four research papers, each addressing the two research questions. The chapter summarizes how the four research papers contribute to the two research questions and how their results complement each other. Furthermore, it elaborates how the research papers support each other by their differences in theory, empirical data, and foci.

Table 2 provides the publication details for each paper, which are included in the appendix. The Approach column shows what research approach was used in each paper. The RQ1 Focus column shows the focus on understanding management of distributed software projects in the papers. The RQ2 Contribution column shows the papers' contribution to support management of distributed software projects.

#	Title	Authors	Publication	Approach	RQ1 Focus	RQ2 Contribution
1	Managing Risks in Distributed Software Projects: An Integrative Framework	Persson JS, Mathiassen L, Boeg J, Madsen TS, and Steinson F	IEEE Transactions on Engineering Management, Vol. 56, No. 3, 2009.	Literature study	Risks	Risk management framework
2	A Process for Managing Risks in Virtual Teams	Persson JS and Mathiassen L	Accepted to IEEE Software	Literature study	Process	Risk management process
3	Enacting Control through Media and Context in Agile Distributed Software Development	Persson JS, Mathiassen L, and Aaen I	First version presented at 14th Americas Conference on Information Systems 2008, Toronto, Canada. Revised version submitted to Information Systems Journal.	Case study	Control	Control practices
4	Collective Minding in Virtual Teams: Investigating Heedful Interrelating with Multimodal Communication	Persson JS and Mathiassen L	Submitted to Information Systems Research	Case study	Coordination	Collective minding framework

Table 2 Research papers

Research paper one and two had an initial main emphasis on RQ2 by investigating how we can support the management of distributed software projects. This was done by developing a framework of what we know about risks and risk resolution techniques through a systematic review of the literature on distributed software projects. Subsequent implementation of a web-based tool helped refine the framework based on empirical evaluation of its practical usefulness. Based on the resulting framework in paper one, a risk management process compliant with CMMI (2006) was developed and illustrated in paper two. These papers therefore provide a risk management artifact for supporting management of distributed software projects. However, the first two papers not only address RQ2, they also provide new understandings that respond to RQ1. The first paper provides an understanding of distributed software projects as a new organization of work, causing a large amount of new risks compared to collocated software projects. In addition, paper two provides an understanding of distributed software projects as a phenomenon that can be managed using generic processes.

Paper three and four had an initial main emphasis on RQ1 by investigating how we can understand the management of distributed software projects. The papers present an investigation of the fundamental software project challenges of control and coordination through a case study. In response to RQ1, the third paper suggests an understanding of management in agile distributed software projects based on the control theory. The investigation suggests that management in distributed software project can be heavily reliant on both formal and informal controls and in both context and practice. In response to RQ2, these findings suggest that the successful management of distributed software projects can be supported by agile principles characterized by informal control. In response to RQ1, paper four suggests an understanding of management practice in agile distributed software projects based on the coordination theory. The investigation suggests that coordination practices with multimodal communication technologies in agile distributed software projects, can exhibit collective minding. In response to RQ2, the management of distributed software projects can be supported by using collective minding as a lens for evaluating coordination performance. In particular, it can shed light on how the use of multimodal communication media can influence performance in a distributed software project.

5.1 Managing Risks in Distributed Software Projects: An Integrative Framework

This paper primarily addresses RQ2 by providing the framework to a risk management artifact for supporting the management of distributed software projects. The paper reports on a systematic review of the literature on geographically distributed software projects. Based on this review, an integrative framework for managing risks in distributed contexts was developed. The integrative framework was subsequently implemented as a web-based tool and refined through empirical evaluation of its practical usefulness.

The paper's research objective was to "*integrate the existing knowledge into a practically useful framework for managing risks inherent in geographically distributed software projects*". The research objective was pursued through two activities.

First, current knowledge was synthesized into conceptualizations of risks and resolution techniques and integrated into a risk-action list framework (Boehm 1991; Iversen et al. 2004) for geographically distributed software projects. This framework consisted of three primary elements: risk assessment, risk resolution, and risk management planning, and it provided heuristics for applying resolution techniques to risk areas. The study developed clear and distinct conceptualizations of both risks and resolution techniques based on existing literature. This was done by:

1) Using available surveys of challenges in distributed software projects to identify relevant risk areas consistent with the state-of-the-art knowledge;
2) Using Leavitt's (1964) systems model, as proposed by Lyytinen et al. (1998), to clarify the primary focus of the identified risk areas;
3) Aggregating a complete list of risk factors identified in the literature and categorizing them according to the proposed risk areas;
4) Characterizing each risk area through detailed definitions of the involved risk factors;
5) Using available surveys of best practices in distributed software projects to identify relevant risk resolution techniques,

6) Categorizing the risk resolution techniques by adopting McFarlan's (1981) well-established categories of project management techniques to distributed contexts, and finally by;
7) Identifying heuristics for applying resolution techniques to risk areas based on the state-of-the-art research on distributed software projects.

Second, the framework was revised iteratively, based on its practical usefulness as evaluated by experienced practitioners. To that end, a variety of empirical evaluation methods were adopted in five evaluations. Initially, software practitioners assessed the relevance and understandability of both risk and risk resolution concepts through two focus groups. Three subsequent evaluations aimed at assessing the practical uses of the framework through workshops with experienced practitioners. The last two workshops were supported by implementing a Web-based tool, making the framework readily available for practical use. In conclusion, the framework was found to be easy to use and to provide relevant support for managing the projects under consideration.

5.2 A Process for Managing Risks in Distributed Teams

This paper primarily addresses RQ2 by providing the process description to a risk management artifact for supporting the management of distributed software projects. The paper reports on a process for managing risks in distributed teams based on the integrative framework presented in paper one. Following CMMI's (2006) generic approach to risk management, the process offers a series of rigorous steps that are readily understood and easy to follow. The process is illustrated with a large-scale, strategic software project with multiple subprojects that crosses organizational and national boundaries. Furthermore, the process is supplemented by suggestions for tailoring the process to a particular project or organization.

In accordance with the CMMI (2006) risk management practice goals the process was structured into three steps: 1) identify and analyze risks, 2) develop risk mitigation plans, and 3) implement risk mitigation plans. The first step was supported by providing identified and categorized distributed team risks, based on paper one. In this step, team members evaluated risk probabilities and impacts and prioritized how to address them. The second step was the development of a risk mitigation plan. This step was supported by the list of resolution techniques and guidelines on how to apply these to address specific risk areas. In the final step, participants related to project objectives and decided on practical approaches. To do so, they considered responsibilities, resources, deliverables, and milestones as key elements in implementing risk mitigation plans. In the paper, each process step was further explained with an illustration, based on data from a distributed software project.

In conclusion, the paper provided suggestions for tailoring the risk management process. This tailoring can be done by considering how to:

1) Keep it simple, to avoid participants seeing the process as an unrewarding administrative burden;
2) Balance participation, by involving the appropriate amount of participants in accordance with the specific project setting and time;
3) Adapt taxonomies, by including only the content fitting the specific project or organization;
4) Integrate plans, by ensuring that the risk management process adheres to project management practices at large;

5) Attract capabilities, in order to reduce proactively the probability and impact of distributed team risks.

5.3 Enacting Control through Media and Context in Agile Distributed Software Development

This paper primarily addresses RQ1 in understanding the management of distributed software projects by providing insights into control practices. The paper reports from an in-depth case study of an agile distributed software project. The study offers an analysis of how control was enacted through the project context and in the mediated communication between project participants. The analysis reveals two important insights contrary to insights from previous research. First, informal control mechanisms are used even though the team is short-lived and rarely meets face-to-face. Second, informal roles and relationships such as clan-like control inherent in agile development are pervasively observed in this distributed setting.

The paper's research question was: how do successful agile distributed software projects enact different elements of control through mediated communication and project context? This research question was pursued with a case study of the *Comapping* project - an agile distributed software project involving participants in Russia and Denmark. Based on interviews and e-conference observations, Kirsch's (2004) elements of control were investigated in the project. This investigation revealed evidence of all of Kirsch's (2004) elements of control in the participant statements about the project context and in the project's e-conferences. Moreover, considerable elements of both informal and formal control were enacted in the e-conference.

Previous research generally questions whether informal control mechanisms can be used when teams are short-lived and rarely meet face-to-face (Powell et al. 2004). In contrast, this paper presented a team, distributed across geography, culture, and companies, developing a successful product while being highly reliant on informal controls. Previous research also suggests that informal roles and relationships, such as clan-like control inherent in agile development, will likely be more difficult to practice in a distributed setting (Harris et al. 2006). In contrast, the results presented in this paper show that agile practices and clan-like controls were pervasively observable in real-time mediated exchanges between project participants and in their accounts of the project context.

5.4 Collective Minding in Virtual Teams: Investigating Heedful Interrelating with Multimodal Communication

This paper primarily addresses RQ1 in order to understand the management of distributed software projects by providing insights into coordination practices. The paper reports on a detailed study of coordination with multimodal communication technology in a virtual team. For this purpose, the paper presents a framework for investigating how multimodal communication technology can support virtual team performance. The framework was used to analyze the patterns of mediated multimodal interactions that allow the team to do collective minding and successfully coordinate their efforts.

The paper's research question was: how does multimodal communication, based on teleconferencing and real-time collaborative modeling, affect collective minding and coordination in virtual teams? This research question was pursued through a quantitative and qualitative analysis of the e-conferences in the *Comapping* project. This analysis identified two types of activities. These were manipulation acts by real-time collaborative modeling and articulation acts by teleconferencing, based on Searle's (1969; 1979) typology of speech acts in cooperative work (Winograd 1987). The actors' capability to interrelate these activities was analyzed by identifying communication breakdowns (Bjørn & Ngwenyama 2009; Ngwenyama 1998). These three analysis constructs were used in relation to the collective mind conceptualization (Weick and Roberts 1993).

The paper provides two key contributions. First, that multimodal communication can facilitate collective minding in virtual teams and positively impact coordination performance. Secondly, that the impact of multimodal communication on virtual team performance can be investigated through an analysis of articulation acts, manipulation acts, and communication breakdowns. The paper presents five propositions based on the study:

1) Multimodal communication, based on teleconferencing and real-time collaborative modeling, can facilitate collective minding in virtual teams and positively impact on coordination performance;
2) The combination of teleconferencing and real-time collaborative modeling can help actors heedfully sense and respond to work-process breakdowns in virtual team coordination;
3) Specification, prioritization, and time boxing of tasks through multimodal communication can facilitate collective minding in virtual teams;
4) Collective minding helps virtual team members overcome differences in cultural and professional knowledge across sites without explicitly sharing that knowledge;
5) Collective minding in virtual teams, based on multimodal communication, can be investigated through a combination of quantitative and qualitative analyses of articulation acts, manipulation acts, and communication breakdowns.

6 Discussion

This chapter presents a discussion of this PhD study's contributions, implications, limitations, and future research. The PhD study's contributions are summarized by answering the research questions. These answers are then related to previous research and management practice in distributed software projects. The limitations of the answers and how they were developed, is also summarized along with suggestions for future research.

6.1 Contributions

This PhD study raises two research questions. How can we understand (RQ1) and how can we support (RQ2) the management of distributed software projects? These two research questions were pursued by two research efforts: A literature study and a case study. The literature study addressed the two research questions grounded in a traditional model-driven perspective and was based on classical systems development research. The case study addressed the two research questions grounded in the emergent agile perspective and was based on an investigation of practices in a distributed software project. These two research efforts were presented in four research papers. The first two papers were based on the literature study and primarily addressed RQ1, with conceptualizations and RQ2, with an artifact (Papers 1 and 2). The second two papers were based on the case study and primarily addressed RQ1, with insights and RQ2, with propositions (Papers 3 and 4), see Table 1.

We can understand distributed software projects as examples of an organization type, which exacerbates existing, and provides new, management challenges (RQ1: Papers 1 and 2). These challenges can be conceptually summarized in eight risk areas: 1) task distribution; 2) knowledge management; 3) geographical distribution; 4) collaboration structure; 5) cultural distribution; 6) stakeholder relations; 7) communication infrastructure; and 8) technology setup (Paper 1). These risk areas encompass existing categorizations of challenges in state-of-the-art literature on distributed projects (Bell & Kozlowski 2002; Herbsleb & Moitra 2001; Powell et al. 2004; Sakthivel 2005; Townsend et al. 1998) and balanced attention to the major dimensions of organizational change (Leavitt 1964; Lyytinen et al. 1998). The identified risk areas are manageable through various forms of planning, control, social integration, and technical integration (McFarlan 1981). The approach for managing these risk areas can therefore, be defined by unifying principles, which operate in many different systemic contexts (Paper 1). The recognition of such unifying principles is what underlies a process understanding. Management of distributed software projects can therefore be understood and defined as a process challenge (Paper 2). In general, this understanding also complies with the traditional model-driven understanding of software project management also underlying CMMI (CMMI-Product-Team 2006).

We can alternatively understand the distribution of software projects as a key facilitator of new business opportunities (RQ1: Papers 3 and 4). In contrast to the risk and process description, where focus is on threats in different organizational settings, distributed software projects can be understood as opportunities facilitated by new practices. This understanding suggests fewer and more socially oriented unifying principles, which should operate in the many different systemic contexts. In other words, management of distributed software projects can be understood as less explicitly definable compared to the previous model-driven understanding. This understanding is compliant with the agile development methodologies that are increasingly popular in distributed software projects (Agerfalk & Fitzgerald 2006; Pries-Heje et al. 2005). Two insights underlying this understanding have been developed from the case study. First, we know that agile characteristics of lightweight processes, ongoing negotiation, and reliance on skilled people introduce challenges related to balancing people- versus process-oriented control in distributed software projects (Ramesh et al. 2006). Furthermore, previous research questions the possibility of successfully adopting informal controls in distributed settings (Harris et al. 2006; Powell et al. 2004). The case study however, showed that successful distributed software projects can rely extensively on informal controls (Paper 3). This insight suggests that the classical management issue, organizational control, can successfully and extensively rely on social and people strategies; this particularly occurred with informal roles and relationships, where participants in the case project acted as both controller and controllee. This is clearly in contrast to the formal evaluation and measurement of behaviors and outcomes suggested in the traditional model-driven understanding underlying CMMI (CMMI-Product-Team 2006). Second, we know that agile methodology emphasizes high interdependency and integration, which has proved very difficult in distributed settings with high reliance on ICT (Gibson & Gibbs 2006). The case study however, showed that collective minding can be observed in successful distributed software projects' multimodal communication (Paper 4). This insight suggests that management of distributed software projects can be understood as a socio-cognitive phenomenon (Fiore et al. 2003; Yoo & Kanawattanachai 2001). This is an understanding that emphasizes collective capabilities in managing distributed software projects. These collective capabilities are developed through heedful interrelating (Weick & Roberts 1993) among participants in the distributed software project's social system. This emphasis on management through social system characteristics is contrasted by the formal organizational control perspective. The organizational control perspective encompasses more emphasis on individual contributions as either controller or controlee (Kirsch & Cummings 1996; Kirsch et al. 2002), compared to the collective minding perspective. The collective minding perspective is therefore, even more than the informal control perspective, significantly different from the traditional model-driven understanding underlying CMMI.

In response to RQ2, the challenge of managing distributed software projects can be supported with a risk management artifact (Papers 1 and 2). The artifact, developed in this PhD study, is an information system which encompass 1) a risk management framework developed from state-of-the-art literature on distributed software projects and empirical evaluations, 2) a web-based tool instantiation of the framework, which facilitated further empirical evaluations of practical usefulness (Paper 1), and 3) a CMMI-compliant (2006) process for using the risk management framework and web-based tool (Paper 2). According to the evaluations by experienced practitioners, the artifact can provide significant support in managing distributed software projects. However, the subparts of the artifact can alternatively be viewed as individual propositions for supporting the management of distributed software projects. These subparts are the risk conceptualizations, the heuristics between risks and resolutions, or the process guidelines. Furthermore, the artifact provides participants in distributed software projects with a vocabulary they can use to identify, reflect on, and share management problems and solutions (Boland 2002; Schön 1983). Hence, the artifact is addressing the need for comprehensive approaches to effectively manage the challenges in distributed projects (Evaristo et al. 2004; Powell et al. 2004)

In an alternative response to RQ2, focus is on propositions for supporting management of distributed software projects (Papers 3 and 4). The propositions are developed through a study of a successful distributed software project. In this particular project, there was a high reliance on agile principles and informal control, inherent in agile methodology. In light of this observation, it can be proposed that agile principles can provide the foundation for the successful management of distributed software projects. This is a proposition supported by previous research on agile distributed software projects (Armour 2007; Holmström et al.

2006; Paasivaara et al. 2008; Ramesh et al. 2006; Sutherland et al. 2007). In contrast to previous propositions on control and distributed projects (Harris et al. 2006; Powell et al. 2004), project success can be achieved more specifically by developing an effective portfolio of controls with high attention to the enactment of informal controls (Paper 3). The second proposition was based on an analysis of the project's ability to coordinate effectively with ICT, which is identified as a key challenge in distributed software projects (Espinosa et al. 2007; Herbsleb & Mockus 2003; Kommeren & Parviainen 2007). This analysis showed that multimodal communication can facilitate collective minding in distributed software projects and positively impact coordination performance (Paper 4). The study also showed that the impact of multimodal communication on distributed software projects' performance can be investigated through a combination of quantitative and qualitative analyses of manipulation acts, articulation acts (Winograd 1987), and communication breakdowns (Bjørn & Ngwenyama 2009). On the basis of this study, collective minding is proposed as a lens to support reflective management practice in distributed software projects. In particular, this proposed reflection is related to the use of multimodal communication media and coordination performance. Distributed projects are highly reliant on ICT (Gibson & Gibbs 2006) and recognizing how the use of available technologies influences performance is critical.

6.2 Implications

This PhD study has several implications for both research and practice. First, the PhD study provided a risk management artifact, encompassing a comprehensive framework for understanding and supporting the management of distributed software projects. This risk management artifact addresses the need for a comprehensive approach to effectively manage the challenges in distributed software projects (Evaristo et al. 2004; Powell et al. 2004). In addition, the process, inherent in the artifact, is compliant with the widespread CMMI model for development v.1.2 (CMMI-Product-Team 2006). While understanding software project management as a generic process issue is widespread among many researchers and practitioners, there exists an increasingly popular opposing understanding. This alternative is the agile understanding of software project management, which is now appearing in distributed software projects (Agerfalk & Fitzgerald 2006; Armour 2007; Holmström et al. 2006; Ramesh et al. 2006). In that context, this PhD's case study addressed the calls for empirical investigation of agile practices (Agerfalk & Fitzgerald 2006; Dybå & Dingsøyr 2008; Lee et al. 2006). In addition, the case study also addressed the calls for research about control (Kirsch 2004; Powell et al. 2004; Yadav et al. 2009) and socio-cognitive reliant coordination in a distributed environment (Fiore et al. 2003; Yoo & Kanawattanachai 2001).

The case study results contradict previous research, questioning whether informal control mechanisms, in general, can be used when teams are short-lived and rarely meet face-to-face (Powell et al. 2004). Furthermore, the case study shows that agile practices and clan-like controls can be pervasively present in a successful distributed software project. This finding underpins the idea that agile methodologies can be adopted successfully in distributed software projects. This contradicts research, which argues that clan control can be increasingly difficult when interaction is mediated using technology and when participants come from different organizations (Harris et al. 2006). The case study results also suggest that the impact of multimodal communication on virtual team performance can be investigated by analyzing articulation acts, manipulation acts, and communication breakdowns. Furthermore, this investigation supports Cooren's (2004) claim that successful collective minding can be explored in interactional patterns between people.

This PhD study has finally provided several contributions relevant to management practice in distributed software projects:

- A risk management framework illustrated with a web-based tool ready for practical use.
- A CMMI-compliant process to help using the risk management framework in practice.
- An illustration of how agile principles and informal controls can be adopted in successful distributed software development.
- An approach for investigating the use of multimodal communication technology in distributed software projects to understand and support coordination performance.

6.3 Limitations

The ways of understanding the management of distributed software projects presented in this PhD study, have limitations. One limitation is that these understandings are not exhaustive in terms of either detail or amount of available understandings of the investigated phenomenon. The understanding from the literature study is limited as it draws on research from specific outlets published over a limited period. Significant developments in understanding the management of distributed software projects may be present in outlets other than the ranked *management* and *management information systems* journals. The case-study-based understanding is grounded in a specific social setting. Transferring this understanding to other distributed software projects requires careful consideration of any differences in the social systems. The social system, from which the understanding was developed, can be significantly different to other distributed software projects' social systems. Additionally, the understandings are grounded in, and likely biased towards, North American and European research and practice discourses. This is an important limitation, because distributed software projects are often geographically distributed across sites that are outside these regions.

Also, the understandings along with the proposed support for managing distributed software projects have undergone limited evaluation. While both the proposed understanding and support has been grounded in practice, their generalizability can be questioned. The pluralistic approach, adopted in this PhD study, emphasizes extensive scrutiny through different research approaches in order to achieve high reliability. The two individual research efforts carried out in parallel provided different understandings and support for managing distributed software projects. While the two different understandings supplement each other, by contrast, a further understanding is still needed to identify to what extent the two can be combined. The adopted research approaches may also have influenced the resulting understandings. This concern is motivated by the two research efforts' differing outcomes in understanding and support. With this backdrop, there is a need to scrutinize available understandings and support for managing distributed software projects using different research approaches. While such extensive scrutiny was not feasible in this PhD study, it is an important direction for future research.

Finally, the researcher's role includes decisions, which can be influenced by personal views and attitudes. These influencing elements may have affected the researcher's choice of inquiry, interpretation of data, and selection of reference points in the body of knowledge. In general, the researcher's mere presence and engagement with informants, as either observer or interviewer, has a significant influence that needs to be taken into consideration. This PhD study is an interpretive research effort and should be appreciated as such.

6.4 Future Research

This PhD study has provided insights into how we can understand and support the management of distributed software projects. These insights have limitations and further scrutiny is needed, with the help of different research approaches, theoretical lenses, and data-documenting practice. In particular, more observations and future research are needed to appreciate how different distributed software projects understand and support management in practice.

The artifact and propositions developed in this PhD study are not sufficient to address the complex challenges in managing distributed software projects. Research and practitioner attention on how we can support this type of project has been voluminous in the past several years. These endeavors appear to be even more important in the future. However, in this pursuit to examine how to support these projects, in-depth investigations of adoptions in real distributed settings are very important. These in-depth investigations of support initiatives in real distributed software projects are needed to create new developments in ICT support. In addition, investigations on the impact of new processes or principles for management in practice, are also equally important avenues for future research.

There is a particular need for future research to emphasize collective capability in the understanding of distributed software projects. This is motivated by the rising popularity of agile methodologies in both collocated and distributed projects, as they often emphasize *collective capabilities*. This understanding is significantly understudied compared with research into the model-driven understanding of software project management, which underlies CMMI (CMMI-Product-Team 2006). Model-driven understanding has been widely recognized and researched for several decades. Therefore, future research needs to prioritize further investigations of emergent understandings. In addition, investigations are needed to examine how emergent understandings can be combined with more established understandings. Also, the argument for pluralism in research (Mingers 2001; 2003) can definitely be valid for understanding practice in distributed software projects, as shown in this PhD study. Future investigations of collective capabilities in distributed software projects can be pursued with the help of conceptualizations, such as *collective mind* (Crowston & Kammerer 1998; Weick & Roberts 1993; Yoo & Kanawattanachai 2001), *shared cognition* (Cannon-Bowers & Salas 2001; Cannon-Bowers et al. 1993; Ensley & Pearce 2001), *distributed cognition* (Boland et al. 1994; Rogers & Ellis 1994; Wright et al. 2000), or *transactive memory systems* (Akgün et al. 2005; Brandon & Hollingshead 2004; Oshri et al. 2008).

This PhD study has, through *interpretation* and *design* activities, pursued knowledge about how we can understand and support the management of distributed software projects. These activities and types of knowledge are only two out three activities in collaborative practice research (Mathiassen 2002). The third activity is *intervention* for developing knowledge of what it actually takes to improve practice. To some extent, this PhD study has intervened in practice, but it has not been the primary focus of either of the two overall research efforts. Action research efforts, extending this PhD study, will therefore, form an important direction for future research. In fact, this PhD study's investigation of the literature on distributed software projects indicates that very little action research has been carried out in this area. This lack of action research also indicates that we know very little of what it actually takes to improve practice in distributed software projects. Therefore, it is suggested that action research investigations are the most important direction for future research.

7 References

Agerfalk, P.J., and Fitzgerald, B. "Flexible and distributed software processes: Old petunias in new bowls?," *Communications of the ACM* (49:10), Oct 2006, pp 26-34.

Akgün, A.E., Byrne, J., Keskin, H., Lynn, G.S., and Imamoglu, S.Z. "Knowledge networks in new product development projects: A transactive memory perspective," *Information & Management* (42:8), 2005, pp 1105-1120.

Alge, B.J., Wiethoff, C., and Klein, H.J. "When does the medium matter? Knowledge-building experiences and opportunities in decision-making teams," *Organizational Behavior and Human Decision Processes* (91:1), May 2003, pp 26-37.

Alter, S., and Ginzberg, M. "Managing uncertainty in MIS implementation," *Sloan Management Review* (20:1), 1978, pp 23-31.

Armour, P.G. "Agile ... and offshore," *Communications of the ACM* (50:1), Jan 2007, pp 13-16.

Barki, H., Rivard, S., and Talbot, J. "Toward an assessment of software development risk," *Journal of Management Information Systems* (10:2), 1993, pp 203-225.

Battin, R.D., Crocker, R., Kreidler, J., and Subramanian, K. "Leveraging resources in global software development," *IEEE Software* (18:2), Mar-Apr 2001, pp 70-77.

Bell, B.S., and Kozlowski, S.W.J. "A typology of virtual teams - implications for effective leadership," *Group & Organization Management* (27:1), Mar 2002, pp 14-49.

Benbasat, I., Goldstein, D.K., and Mead, M. "The case research strategy in studies of information systems," *MIS Quarterly* (11:3), 1987, pp 369-386.

Bergiel, B.J., Bergiel, E.B., and Balsmeier, P.W. "Nature of virtual teams: a summary of their advantages and disadvantages," *Management Research News* (31:2), 2008, pp 99-110.

Berry, D.M. "The Software engineering silver bullet conundrum," *IEEE Software* (25:2), 2008, pp 18-19.

Bjørn, P., and Ngwenyama, O. "Virtual team collaboration: building shared meaning, resolving breakdowns and creating translucence," *Information Systems Journal* (19:3), 2009, pp 227-253.

Bleecker, S.E. "The virtual organization," *Futurist* (28:2), Mar-Apr 1994, pp 9-14.

Boehm, B.W. "Software risk management - principles and practices," *IEEE Software* (8:1), Jan 1991, pp 32-41.

Boland, R.J. "Design in the punctuation of management action," in: *Managing as Designing: Creating a Vocabulary for Management Education and Research,* R.J. Boland (ed.), Frontiers of Management Workshop, Weatherhead School of Management (available online at http://design.cwru.edu), 2002.

Boland, R.J., Tenkasi, R.V., and Teeni, D. "Designing information technology to support distributed cognition," *Organization Science* (5:3), 1994, pp 456-475.

Brandon, D.P., and Hollingshead, A.B. "Transactive memory systems in organizations: matching tasks, expertise, and people," *Organization Science* (15:6), 2004, pp 633-644.

Breu, K., and Hemingway, C.J. "Making organisations virtual: the hidden cost of distributed teams," *Journal of Information Technology* (19:3), Sep 2004, pp 191-202.

Braa, K., and Vidgen, R. "Interpretation, intervention, and reduction in the organizational laboratory: a framework for in-context information system research," *Accounting, Management and Information Technologies* (9:1), 1999, pp 25-47.

Cannon-Bowers, J.A., and Salas, E. "Reflections on shared cognition," *Journal of Organizational Behavior* (22:2), 2001, pp 195-202.

Cannon-Bowers, J.A., Salas, E., and Converse, S. "Shared mental models in expert team decision making," in: *Individual and Group Decision Making,* N.J. Castellan (ed.), Lawrence Erbaum Associates, Hillside, NJ, 1993, pp. 221-243.

Cardinal, L.B., Sitkin, S.B., and Long, C.P. "Balancing and rebalancing in the creation and evolution of organizational control," *Organization Science* (15:4) 2004, pp 411-431.

Carmel, E., and Agarwal, R. "Tactical approaches for alleviating distance in global software development," *IEEE Software* (18:2), Mar-Apr 2001, pp 22-29.

Cascio, W.F., and Shurygailo, S. "E-leadership and virtual teams," *Organizational Dynamics* (31:4), 2003, pp 362-376.

Checkland, P., and Holwell, S. "Action research: its nature and validity," *Systemic Practice and Action Research* (11:1), 1998, pp 9-21.

Choudhury, V., and Sabherwal, R. "Portfolios of control in outsourced software development projects," *Information Systems Research* (14:3), 2003, pp 291-314.

Chudoba, K.M., Wynn, E., Lu, M., and Watson-Manheim, M.B. "How virtual are we? Measuring virtuality and understanding its impact in a global organization," *Information Systems Journal* (15:4), Oct 2005, pp 279-306.

CMMI-Product-Team "CMMI for Development, Version 1.2," Carnegie Mellon Software Engineering Institute, Pittsburg, PA, 2006.

Conchúir, E.Ó., Olsson, H.H., Ågerfalk, P.J., and Fitzgerald, B. "Benefits of global software development: exploring the unexplored," *Software Process: Improvement and Practice (In Press)*), 2009.

Cooren, F. "The communicative achievement of collective minding: analysis of board meeting excerpts," *Management Communication Quarterly* (17:4), 2004, pp 517-551.

Crowston, K., and Kammerer, E.E. "Coordination and collective mind in software requirements development," *IBM Systems Journal* (37:2), 1998, pp 227-245.

Curseu, P.L., Schalk, R., and Wessel, I. "How do virtual teams process information? A literature review and implications for management," *Journal of Managerial Psychology* (23:6), 2008, pp 628-652.

Damian, D., and Moitra, D. "Global software development: How far have we come," *IEEE Software* (23:5), 2006, pp 17–19.

Damian, D., Sengupta, B., and Lanubile, F. "Global software development: where are we headed?," *Software Process: Improvement and Practice* (13:6), 2008, pp 473-475.

Davis, G.B. "Strategies for information requirements determination," *IBM Systems Journal* (21:1), 1982, pp 4-30.

DeRosa, D.M., Hantula, D.A., Kock, N., and D'Arcy, J. "Trust and leadership in virtual teamwork: a media naturalness perspective," *Human Resource Management* (43:2-3), Sum 2004, pp 219-232.

Dibbern, J., Goles, T., Hirschheim, R., and Jayatilaka, B. "Information systems outsourcing: a survey and analysis of the literature," *The DATA BASE for Advances in Information Systems* (35:4), 2004, pp 6-102.

Donaldson, S.E., and Siegel, S.G. *Successful software development* Prentice Hall, Upper Saddle River, NJ, 2001.

Dubé, L., and Paré, G. "Global virtual teams," *Communications of the ACM* (44:12), Dec 2001, pp 71-73.

Dybå, T., and Dingsøyr, T. "Empirical studies of agile software development: a systematic review," *Information and Software Technology* (50:9-10), 2008, pp 833-859.

Ebert, C., and De Neve, P. "Surviving global software development," *IEEE Software* (18:2), 2001, pp 62-69.

Ebert, C., Murthy, B.K., and Jha, N.N. "Managing risks in global software engineering: principles and practices," 3rd IEEE International Conference on Global Software Engineering, IEEE Computer Soc, Bangalore, INDIA, 2008, pp. 131-140.

Eisenhardt, K.M. "Control - organizational and economic approaches," *Management Science* (31:2), 1985, pp 134-149.

Ensley, M.D., and Pearce, C.L. "Shared cognition in top management teams: implications for new venture performance," *Journal of Organizational Behavior* (22:2), 2001, pp 145-160.

Erickson, J.M., and Evaristo, R. "Risk factors in distributed projects," Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006.

Espinosa, J.A., DeLone, W., and Lee, G. "Global boundaries, task processes and IS project success: a field study," *Information Technology and People* (19:4), 2006, pp 345-370.

Espinosa, J.A., Slaughter, S.A., Kraut, R.E., and Herbsleb, J.D. "Team knowledge and coordination in geographically distributed software development," *Journal of Management Information Systems* (24:1), Sum 2007, pp 135-169.

Evaristo, J.R., Scudder, R., Desouza, K.C., and Sato, O. "A dimensional analysis of geographically distributed project teams: a case study," *Journal of Engineering and Technology Management* (21:3), Sep 2004, pp 175-189.

Farmer, M. "DecisionSpace infrastructure: agile development in a large, distributed team," Agile Development Conference, 2004, pp. 95-99.

Farshchian, B.A. "Integrating geographically distributed development teams through increased product awareness," *Information Systems* (26:3), May 2001, pp 123-141.

Fiore, S.M., Salas, E., Cuevas, H.M., and Bowers, C.A. "Distributed coordination space: toward a theory of distributed team process and performance," *Theoretical Issues in Ergonomics Science* (4:3/4), 2003, pp 340-364.

Franke, R.H., Edlund, T.W., and Oster, F. "The Development of Strategic Management - Journal Quality and Article Impact," *Strategic Management Journal* (11:3), Mar-Apr 1990, pp 243-253.

Furst, S., Blackburn, R., and Rosen, B. "Virtual team effectiveness: a proposed research agenda," *Information Systems Journal* (9:4), Oct 1999, pp 249-269.

Galanter, M. *Cults* Oxford University Press, New York, 1989.

Gao, J.Z., Chen, C., Toyoshima, Y., and Leung, D.K. "Engineering on the Internet for global software production," *Computer* (32:5), May 1999, pp 38-47.

Gibson, C.B., and Gibbs, J.L. "Unpacking the concept of virtuality: the effects of geographic dispersion, electronic dependence, dynamic structure, and national diversity on team innovation," *Administrative Science Quarterly* (51:3), 2006, pp 451-495.

Gillam, C., and Oppenheim, C. "Review article: reviewing the impact of virtual teams in the information age," *Journal of Information Science* (32:2), 2006, pp 160-175.

Gomezmejia, L.R., and Balkin, D.B. "Determinants of Faculty pay - an agency theory perspective," *Academy of Management Journal* (35:5), Dec 1992, pp 921-955.

Gopal, A., Mukhopadhyay, T., and Krishnan, M.S. "The role of software processes and communication in offshore software development," *Communications of the ACM* (45:4), 2002, pp 193-200.

Harris, M., Hevner, A.R., and Collins, R.W. "Controls in flexible software development," Proceedings of the 39th Annual Hawaii International Conference on System Sciences, IEEE, Hawaii, 2006.

Herbsleb, J.D., and Grinter, R.E. "Architectures, coordination, and distance: Conway's law and beyond," *IEEE Software* (16:5), Sep-Oct 1999, pp 63-70.

Herbsleb, J.D., and Mockus, A. "An empirical study of speed and communication in globally distributed software development," *IEEE Transactions on Software Engineering* (29:6), Jun 2003, pp 481-494.

Herbsleb, J.D., and Moitra, D. "Global software development," *IEEE Software* (18:2), Mar-Apr 2001, pp 16-20.

Hertel, G., Geister, S., and Konradt, U. "Managing Virtual teams: a review of current empirical research," *Human Resource Management Review* (15:1), 2005, pp 69-95.

Hevner, A.R., March, S.T., Park, J., and Ram, S. "Design science in information systems research," *MIS Quarterly* (28:1), 2004, pp 75-106.

Holmström, H., Fitzgerald, B., Ågerfalk, P.J., and Conchúir, E.Ó. "Agile practices reduce distance in global software development," *Information Systems Management* (23:3), 2006, pp 7-18.

Hwang, S. "Utilizing qualitative data analysis software: a review of Atlas. ti," *Social Science Computer Review* (26:4), 2008, pp 519-527.

Iacovou, C.L., and Nakatsu, R. "A risk profile of offshore-outsourced development projects," *Communications of the ACM* (51:6), 2008, pp 89-94.

Iversen, J.H., Mathiassen, L., and Nielsen, P.A. "Managing risk in software process improvement: an action research approach," *MIS Quarterly* (28:3), Sep 2004, pp 395-433.

Johnson, J.L., and Podsakoff, P.M. "Journal influence in the field of management - an analysis using Salancik Index in a dependency network," *Academy of Management Journal* (37:5), Oct 1994, pp 1392-1407.

Jones, C. *Assessment and control of software risks* Yourdon Press Prentice Hall, Upper Saddle River, NJ, 1994.

Kaiser, K.M., and Hawk, S. "Evolution of offshore software development: from outsourcing to cosourcing," *MIS Quarterly Executive* (3:2), 2004, pp 69-81.

Kanawattanachai, P., and Yoo, Y. "The Impact of knowledge coordination on virtual team performance over time," *MIS Quarterly* (31:4), 2007, p 783.

Katerattanakul, P., Han, B., and Hong, S.G. "Objective quality ranking of computing journals," *Communications of the ACM* (46:10), Oct 2003, pp 111-114.

Kayworth, T., and Leidner, D. "The global virtual manager: a prescription for success," *European Management Journal* (18:2), April 2000, pp 183-194.

Kayworth, T.R., and Leidner, D.E. "Leadership effectiveness in global virtual teams," *Journal of Management Information Systems* (18:3), Win 2001, pp 7-40.

Keil, M., Cule, P.E., Lyytinen, K., and Schmidt, R.C. "A framework for identifying software project risks," *Communications of the ACM* (41:11), 1998, pp 76-83.

Kirkman, B.L., and Mathieu, J.E. "The dimensions and antecedents of team virtuality," *Journal of Management* (31:5), October 1, 2005, pp 700-718.

Kirkman, B.L., Rosen, B., Tesluk, P.E., and Gibson, C.B. "The impact of team empowerment on virtual team performance: the moderating role of face-to-face interaction," *Academy of Management Journal* (47:2), Apr 2004, pp 175-192.

Kirsch, L.J. "The management of complex tasks in organizations: controlling the systems development process," *Organization Science* (7:1), Jan-Feb 1996, pp 1-21.

Kirsch, L.J. "Portfolios of control modes and IS project management," *Information Systems Research* (8:3), Sep 1997, pp 215-239.

Kirsch, L.J. "Software project management: an integrated perspective for an emerging paradigm," in: *Framing the Domain of IT Management: Projecting the Future Through the Past,* R.W. Zmud (ed.), PinnFlex Education Resources, Cincinnati, OH, 2000, pp. 285–304.

Kirsch, L.J. "Deploying common systems globally: the dynamics of control," *Information Systems Research* (15:4), Dec 2004, pp 374-395.
Kirsch, L.J., and Cummings, L.L. "Contextual influences on self-control of is professionals engaged in systems development," *Accounting, Management and Information Technologies* (6:3), 1996, pp 191-219.
Kirsch, L.J., Sambamurthy, V., Ko, D.G., and Purvis, R.L. "Controlling information systems development projects: the view from the client," *Management Science* (48:4), Apr 2002, pp 484-498.
Kommeren, R., and Parviainen, P. "Philips experiences in global distributed software development," *Empirical Software Engineering* (12:6), 2007, pp 647-660.
Kraut, R., Steinfield, C., Chan, A., Butler, B., and Hoag, A. "Coordination and virtualization: the role of electronic networks and personal relationships," *Journal of Computer-Mediated Communication* (3:4), 1998.
Kraut, R.E., and Streeter, L.A. "Coordination in software development," *Communications of the ACM* (38:3), 1995, pp 69-81.
Krishna, S., Sahay, S., and Walsham, G. "Managing cross-cultural issues in global software outsourcing," *Communications of the ACM* (47:4), Apr 2004, pp 62-66.
Kulpa, M.K., and Johnson, K.A. *Interpreting the CMMI: A Process Improvement Approach* CRC Press, Boca Raton, Florida, 2003, p. 414.
Leavitt, H.J. "Applied organization change in industry: structural, technical, and human approaches," *New Perspectives in Organizational Research* 1964, pp 55-71.
Lee, O.K., Banerjee, P., Lim, K.H., Kumar, K., Van Hillegersberg, J., and Wei, K.K. "Aligning IT components to achieve agility in globally distributed system development," *Communications of the ACM* (49:10), Oct 2006, pp 48-54.
Lewin, A.Y., Massini, S., and Peeters, C. "Why are companies offshoring innovation? The emerging global race for talent," *Journal of International Business Studies (forthcoming)* 2008.
Lowry, P., Romans, D., and Curtis, A. "Global journal prestige and supporting disciplines: A scientometric study of information systems journals," *Journal of the Association for Information Systems* (5:2), February 2004.
Lyytinen, K., Mathiassen, L., and Ropponen, J. "A framework for software risk management," *Journal of Information Technology* (11:4), Dec 1996, pp 275-285.
Lyytinen, K., Mathiassen, L., and Ropponen, J. "Attention shaping and software risk - a categorical analysis of four classical risk management approaches," *Information Systems Research* (9:3), Sep 1998, pp 233-255.
Majchrzak, A., Malhotra, A., and John, R. "Perceived individual collaboration know-how development through information technology-enabled contextualization: evidence from distributed teams," *Information Systems Research* (16:1), Mar 2005, pp 9-27.
Majchrzak, A., Rice, R.E., Malhotra, A., King, N., and Ba, S. "Technology adaptation: the case of a computer-supported inter-organizational virtual team," *MIS Quarterly* (24:4), 2000, pp 569-600.
Malone, T.W., and Crowston, K. "Toward an interdisciplinary theory of coordination - Tech. Rep. no. 120," Center for Coordination Science, Sloan School of Management, Massachusetts Institute of Technology, 1991.
Malone, T.W., and Crowston, K. "The interdisciplinary study of coordination," *ACM Computing Surveys* (26:1), Mar 1994, pp 87-119.
Manning, S., Massini, S., and Lewin, A.Y. "A dynamic perspective on next-generation offshoring: the global sourcing of science and engineering talent," *Academy of Management Perspectives* (22:3), 2008, pp 35-54.

Markus, M.L., Manville, B., and Agres, C.E. "What makes a virtual organization work?," *Sloan Management Review* (42:1), 2000, pp 13-26.

Martins, L.L., Gilson, L.L., and Maynard, M.T. "Virtual teams: what do we know and where do we go from here?," *Journal of Management* (30:6), 2004, pp 805-835.

Massey, A.P., Montoya-Weiss, M.M., and Hung, Y.T. "Because time matters: temporal coordination in global virtual project teams," *Journal of Management Information Systems* (19:4), Spr 2003, pp 129-155.

Mathiassen, L. "Collaborative practice research," *Information Technology and People* (15:4), 2002, pp 321-345.

Mathiassen, L., Munk-Madsen, A., Nielsen, P.A., and Stage, J. *Object-oriented Analysis & Design* Marko Publishing House, Aalborg, Denmark, 2000.

Mathiassen, L., Saarinen, T., Tuunanen, T., and Rossi, M. "A contingency model for requirements development," *Journal of the Association for Information Systems* (8:11), 2007, pp 570-598.

Maznevski, M.L., and Chudoba, K.M. "Bridging space over time: global virtual team dynamics and effectiveness," *Organization Science* (11:5), 2000, pp 473-492.

McFarlan, F.W. "Portfolio approach to information-systems," *Harvard Business Review* (59:5), 1981, pp 142-150.

Meyer, B. "The unspoken revolution in software engineering," *IEEE Computer* (39:1), 2006, pp 121-123.

Mingers, J. "Combining IS research methods: towards a pluralist methodology," *Information Systems Research* (12:3), Sep 2001, pp 240-259.

Mingers, J. "The paucity of multimethod research: a review of the information systems literature," *Information Systems Journal* (13:3), 2003, pp 233-249.

Montoya-Weiss, M.M., Massey, A.P., and Song, M. "Getting it together: temporal coordination and conflict management in global virtual teams," *Academy of Management Journal* (44:6), 2001, pp 1251-1262.

Mortensen, M., and Hinds, P.J. "Conflict and shared identity in geographically distributed teams," *International Journal of Conflict Management* (12:3), 2001, pp 212-238.

Mowshowitz, A. "Virtual organization," *Communications of the ACM* (40:9), 1997, pp 30-37.

Moynihan, T. "An inventory of personal constructs for information systems project risk researchers," *Journal of Information Technology* (11:4), 1996, pp 359-372.

Muhr, T. "ATLAS.ti - The knowledge workbench," *Scientific Software Development*, Berlin, Germany, 2008.

Nakatsu, R.T., and Iacovou, C.L. "A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: a two-panel Delphi study," *Information & Management* (46:1), Jan 2009, pp 57-68.

Narayanaswamy, R., and Henry, R.M. "Effects of culture on control mechanisms in offshore outsourced IT projects," Proceedings of the 2005 ACM SIGMIS CPR conference on computer personnel research, Atlanta, Georgia, USA, 2005, pp. 139-145.

Ngwenyama, O.K. "Groupware, social action and organizational emergence: on the process dynamics of computer mediated distributed work," *Accounting, Management and Information Technologies* (8:2-3), 1998, pp 127-146.

Nicholson, B., and Sahay, S. "Embedded knowledge and offshore software development," *Information and Organization* (14:4), 2004, pp 329-365.

Orsak, G.C., and Etter, D.M. "Connecting the engineer to the 21st century through virtual teaming," *IEEE Transactions on Education* (39:2), May 1996, pp 165-172.

Oshri, I., van Fenema, P., and Kotlarsky, J. "Knowledge transfer in globally distributed teams: the role of transactive memory," *Information Systems Journal* (18:6), 2008, pp 593-616.

Ouchi, W.G. "Transmission of control through organizational hierarchy," *Academy of Management Journal* (21:2), 1978, pp 173-192.

Ouchi, W.G. "Conceptual-framework for the design of organizational control mechanisms," *Management Science* (25:9), 1979, pp 833-848.

Ouchi, W.G. "Markets, bureaucracies, and clans," *Administrative Science Quarterly* (25:1), 1980, pp 129-141.

Ould, M.A. *Managing Software Quality and Business Risk* John Wiley & Sons, Inc., New York, NY, 1999.

Paul, S., Seetharaman, P., Samarah, I., and Mykytyn, P.P. "Impact of heterogeneity and collaborative conflict management style on the performance of synchronous global virtual teams," *Information & Management* (41:3), Jan 2004, pp 303-321.

Pauleen, D.J. "An inductively derived model of leader-initiated relationship building with virtual team members," *Journal of Management Information Systems* (20:3), Win 2003, pp 227-256.

Pauleen, D.J., and Yoong, P. "Relationship building and the use of ICT in boundary-crossing virtual teams: a facilitator's perspective," *Journal of Information Technology* (16:4), Dec 2001, pp 205-220.

Pfannenstein, L.L., and Tsai, R.J. "Offshore outsourcing: current and future effects on American IT industry," *Information Systems Management* (21:4), 2004, pp 72-80.

Piccoli, G., and Ives, B. "Trust and the unintended effects of behavior control in virtual teams," *MIS Quarterly* (27:3), Sep 2003, pp 365-395.

Powell, A., Piccoli, G., and Ives, B. "Virtual teams: a review of current literature and directions for future research," *DATA BASE for Advances in Information Systems* (35:1), Win 2004, pp 6-36.

Pries-Heje, J., Baskerville, R., and Hansen, G.I. "Strategy models for enabling offshore outsourcing: Russian short-cycle-time software development," *Information Technology for Development* (11:1), 2005.

Prikladnicki, R., Evaristo, R., Audy, J.L.N., and Yamaguti, M.H. "Risk management in distributed IT projects: integrating strategic, tactical, and operational levels," *International Journal of e-Collaboration* (2:4), 2006, pp 1-18.

Paasivaara, M., Durasiewicz, S., and Lassenius, C. "Using scrum in a globally distributed project: a case study," *Software Process: Improvement and Practice* (13:6), 2008, pp 527-544.

Rainer, R.K., and Miller, M.D. "Examining differences across journal rankings," *Communications of the ACM* (48:2), Feb 2005, pp 91-94.

Ramesh, B., Cao, L., Mohan, K., and Xu, P. "Can distributed software development be agile?," *Communications of the ACM* (49:10), Oct 2006, pp 41-46.

Ramesh, V., and Dennis, A.R. "The object-oriented team: lessons for virtual teams from global software development," Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 2002, pp. 212-221.

Ripeanu, M., Singh, M.P., and Vazhkudai, S.S. "Virtual organizations [guest editors' introduction]," *IEEE Internet Computing* (12:2), 2008, pp 10-12.

Robey, D., Huoy Min, K., and Powers, C. "Situated learning in cross-functional virtual teams," *IEEE Transactions on Professional Communication* (43:1), 2000, pp 51-66.

Rogers, Y., and Ellis, J. "Distributed Cognition: an alternative framework for analyzing and explaining collaborative working," *Journal of Information Technology* (9:2), 1994, pp 119-128.

Ropponen, J., and Lyytinen, K. "Components of software development risk: how to address them? A project manager survey," *IEEE Transactions on Software Engineering* (26:2), 2000, pp 98-112.
Rutkowski, A.F., Vogel, D.R., Van Genuchten, M., Bemelmans, T.M.A., and Favier, M. "E-collaboration: the reality of virtuality," *IEEE Transactions on Professional Communication* (45:4), 2002, pp 219-230.
Sakthivel, S. "Virtual workgroups in offshore systems development," *Information and Software Technology* (47:5), Mar 2005, pp 305-318.
Sakthivel, S. "Managing risks in offshore systems development," *Communications of the ACM* (50:4), 2007, pp 69-75.
Sarker, S., Nicholson, D.B., and Joshi, K.D. "Knowledge transfer in virtual systems development teams: an exploratory study of four key enablers," *IEEE Transactions on Professional Communication* (48:2), 2005, pp 201-218.
Sarker, S., and Sahay, S. "Implications of space and time for distributed work: an interpretive study of US-Norwegian systems development teams," *European Journal of Information Systems* (13:1), Mar 2004, pp 3-20.
Schiller, S.Z., and Mandviwalla, M. "Virtual team research: an analysis of theory use and a framework for theory appropriation," *Small Group Research* (38:1), February 2007, pp 12-59.
Schön, D.A. *The Reflective Practitioner: How Professionals Think in Action* Basic Books, New York, 1983.
Searle, J.R. *Speech Acts* Cambridge University Press, Cambridge, England, 1969.
Searle, J.R. "A taxonomy of illocutionary acts," in: *Expression and Meaning: Studies in the Theory of Speech Acts*, Cambridge University Press, Cambridge, 1979, pp. 1-29.
Sherif, K., Zmud, R., and Browne, G.J. "Managing peer-to-peer conflicts in disruptive information technology innovations: the case of software reuse," *MIS Quarterly* (30:1), 2006, pp 339-356.
Singh, B. "Interconnected roles (IR): a coordination model - Tech, Rep. CT-84-92.," Microelectronics and Computer Technology Corp., Austin, TX, 1992.
Smite, D., Wohlin, C., Feldt, R., and Gorschek, T. "Reporting empirical research in global software engineering: a classification scheme," IEEE International Conference on Global Software Engineering, 2008, pp. 173-181.
Smith, M.A., Mitra, S., and Narasimhan, S. "Offshore outsourcing of software development and maintenance: a framework for issues," *Information & Management* (31:3), 1996, pp 165-175.
Sole, D., and Edmondson, A. "Situated knowledge and learning in dispersed teams," *British Journal of Management* (13:S2), 2002, pp S17-S34.
Starbuck, W.H., and Milliken, F.J. "Challenger: fine-tuning the odds until something breaks," *Journal of Management Studies* (25:4), 1988, pp 319-340.
Sutherland, J., Viktorov, A., Blount, J., and Puntikov, N. "Distributed scrum: agile project management with outsourced development teams," Proceedings of the 40th Annual Hawaii International Conference on System Sciences, 2007.
Tan, B.C.Y., Wei, K.K., Huang, W.W., and Ng, G.N. "A dialogue technique to enhance electronic communication in virtual teams," *IEEE Transactions on Professional Communication* (43:2), Jun 2000, pp 153-165.
Thomson.Scientific "Web of Science - http://scientific.thomson.com/products/wos/," 2005.
Townsend, A.M., DeMarie, S.M., and Hendrickson, A.R. "Virtual teams: technology and the workplace of the future," *Academy of Management Executive* (12:3), 1998, pp 17-29.

Warkentin, M., and Beranek, P.M. "Training to improve virtual team communication," *Information Systems Journal* (9:4), Oct 1999, pp 271-289.

Warkentin, M.E., Sayeed, L., and Hightower, R. "Virtual teams versus face-to-face teams: an exploratory study of a Web-based conference system," *Decision Sciences* (28:4), Fall 1997, pp 975-996.

Webster, J., and Watson, R.T. "Analyzing the past to prepare for the future: writing a literature review," *MIS Quarterly* (26:2), Jun 2002, pp XIII-XXIII.

Weick, K.E. "Contradictions in a community of scholars: the cohesion-accuracy tradeoff," *Review of Higher Education* (6:4), 1983, pp 253-267.

Weick, K.E., and Roberts, K.H. "Collective mind in organizations - heedful interrelating on flight decks," *Administrative Science Quarterly* (38:3), Sep 1993, pp 357-381.

Weill, P., and Olson, M.H. "An assessment of the contingency theory of management information systems," *Journal of Management Information Systems* (6:1), 1989, pp 59-85.

Whitman, M.E., Hendrickson, A.R., and Townsend, A.M. "Research commentary. Academic rewards for teaching, research, and service: data and discourse," *Information Systems Research* (10:2), Jun 1999, pp 99-109.

Wiesenfeld, B.M., Raghuram, S., and Garud, R. "Communication patterns as determinants of organizational identification in a virtual organization," *Journal of Computer-Mediated Communication* (3:4), 1998.

Winograd, T. "A language/action perspective on the design of cooperative work," *Human-Computer Interaction* (3:1), 1987, pp 3-30.

Winter, R. "Design science research in Europe," *European Journal of Information Systems* (17:5), 2008, pp 470-475.

Wolf, T., Nguyen, T., and Damian, D. "Does distance still matter?," *Software Process: Improvement and Practice* (13:6), 2008, pp 493-510.

Wright, P.C., Fields, R.E., and Harrison, M.D. "Analyzing human-computer interaction as distributed cognition: the resources model," *Human-Computer Interaction* (15:1), 2000, pp 1-41.

Xue, Y.J., Sankar, C.S., and Mbarika, V.W.A. "Information technology outsourcing and virtual team," *Journal of Computer Information Systems* (45:2), Win 2004, pp 9-16.

Yadav, V., Adya, M., Sridhar, V., and Nath, D. "Flexible global software development: antecedents of success in requirements analysis," *Journal of Global Information Management,* (17:1), 2009, pp 1-31.

Yin, R.K. *Case Study Research: Design and Methods* (Third ed.) Sage Publications Inc, Thousand Oaks, California, 2003, p. 200.

Yoo, Y., and Kanawattanachai, P. "Developments of transactive memory systems and collective mind in virtual teams," *The International Journal of Organizational Analysis* (9:2), 2001, pp 187-208.

Zigurs, I. "Leadership in virtual teams: oxymoron or opportunity?," *Organizational Dynamics* (31:4), 2003, pp 339-351.

8 Appendix: The Papers

508 IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, VOL. 56, NO. 3, AUGUST 2009

Managing Risks in Distributed Software Projects: An Integrative Framework

John Stouby Persson, Lars Mathiassen, *Member, IEEE*, Jesper Boeg, Thomas Stenskrog Madsen, and Flemming Steinson

***Abstract*—Software projects are increasingly geographically distributed with limited face-to-face interaction between participants. These projects face particular challenges that need careful managerial attention. While risk management has been adopted with success to address other challenges within software development, there are currently no frameworks available for managing risks related to geographical distribution. On this background, we systematically review the literature on geographically distributed software projects. Based on the review, we synthesize what we know about risks and risk resolution techniques into an integrative framework for managing risks in distributed contexts. Subsequent implementation of a Web-based tool helped us refine the framework based on empirical evaluation of its practical usefulness. We conclude by discussing implications for both research and practice.**

***Index Terms*—Communication and collaboration, distributed software projects, risk management.**

I. Introduction

GLOBAL competition, increased need for flexibility, access to global resources, and substantial financial gains drive companies to engage in geographically distributed software projects (GDSPs) [37], [91]. Moreover, as electronic communication infrastructures are now readily available, geographically distributed projects have become increasingly feasible to organize and manage [97], [101]. However, these projects face numerous management challenges that are inherent to their distributed nature, e.g., limited social interaction [22], [35], [37], [87], language barriers [22], [88], [100], and time zone differences [12], [16], [25], [47], [100]. While the growth in GDSPs has attracted increasing attention in the literature, there is still considerable variation in the terms used, including virtual teams [88], global virtual teams [41], virtual work groups [99], virtual organizations [59], distributed projects [25], and geographically distributed development teams [28]. In this paper, we focus on GDSPs that "consist of geographically dispersed people working interdependently with shared purpose across space, time, and organizational boundaries and using technology to communicate and collaborate" [99].

Manuscript received August 22, 2007; revised March 24, 2008, June 24, 2008, and August 25, 2008. First published April 17, 2009; current version published July 17, 2009. Review of this manuscript was arranged by Department Editor R. Sabherwal.

J. S. Persson is with the Department of Computer Science, Aalborg University, Aalborg 9220, Denmark (e-mail: john@cs.aau.dk).

L. Mathiassen is with Georgia State University, Atlanta, GA 30303 USA (e-mail: lmathiassen@ceprin.org).

J. Boeg is with Trifork A/S, Aarhus DK-8000, Denmark (e-mail: jbo@eos.dk).

T. S. Madsen and F. Steinson are with Systematic Software Engineering A/S, Aarhus DK-8000, Denmark (e-mail: tsg@systematic.dk; flemming@steinson.dk).

Digital Object Identifier 10.1109/TEM.2009.2013827

A portfolio of approaches to alleviate specific challenges in GDSPs has been proposed, e.g., dialogue technique [105], list of best practices [5], [24], and a person–environment fit model [103]. These contributions are valuable, but they do not constitute a comprehensive approach to effectively manage the challenges in GDSPs [25], [91]. Moreover, while risk management has been applied successfully in collocated software development [39], [67] the resulting approaches fail to address the unique communicative and collaborative challenges that distinguish GDSPs from traditional software projects [37]. This research was therefore guided by the overall objective to integrate existing knowledge into a practically useful framework for managing risks inherent in GDSPs. In order to do this, we first reviewed the literature to identify and conceptualize the specific risks inherent in GDSPs and to identify and conceptualize the available resolution techniques [67]. Second, we integrated these insights into a framework for applying risk resolution techniques to risks, implemented a tool for practical use of the framework, and refined the framework based on empirical evaluation of its practical usefulness.

The basic principles of risk management seek to generalize patterns of relations between organizational contexts (in the form of risk areas and underlying risk factors) and use of technologies (in the form of resolution techniques) in ways that support human action [67]. A software risk denotes an aspect of a development task, process, or environment, which, if ignored, increases the likelihood of project failure [67]. Practitioners can assess the degree of risk either quantitatively as the probability of unsatisfactory events multiplied by the loss associated with their outcome, or qualitatively by referring to the uncertainty surrounding the project and the magnitude of potential loss associated with project failure [4]. Risk management helps practitioners assess problematic aspects of a project, emphasizes potential causes of failure, helps link potential threats to possible actions, and facilitates a shared perception of a project among its participants [66], [67]. Risk frameworks and associated tools have previously been successfully developed to identify, analyze, and tackle project portfolio risks [23], [75], software development risks [4], [7], [14], [21], [27], [50], [79], [83], [95], software requirements risks [11], [18], [72], software process improvement risks [39], and implementation risks [3], [48], [57], [63], [65].

Our research draws upon a systematic review of the literature on GDSPs (Section II); synthesizes conceptualizations of risks (Section III) and resolution techniques (Section IV) and integrates these into a framework and related tool for managing risks in distributed contexts (Section V); and finally, documents how the framework and tool were refined based on evaluations of their practical usefulness (Section VI). We conclude by

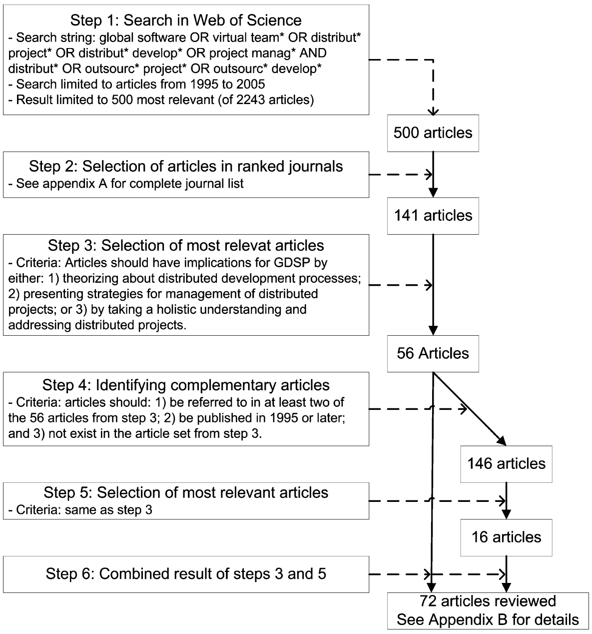

Fig. 1. Literature identification.

discussing the contribution of this research and its implications for theory and practice (Section VII).

II. Literature Review

The primary goal of a literature review is to achieve a complete result focused on concepts [113]. Thus, the two most important tasks are to decide how to identify the relevant literature and how to conceptually structure the analysis [114].

A. Identifying the Literature

As our field of interest was managerial challenges in GDSPs, we chose a wide range of management information systems and management research journals as the primary sources of information. This was based on the assumption that many of the challenges faced by managers of GDSPs are similar to the ones encountered within other industries involved in distributed projects.

Inspired by Webster and Watson [113], we adopted a rigorous approach to identify relevant articles in leading journals. From the identified set of articles, we searched backward by following the used references. This approach was combined with Weill and Olson's [114] suggestion to use structured critique to further steer the selection of articles. Our combined approach is summarized in Fig. 1.

In the first step, we searched for relevant articles in the Web of Science article database. The search was limited to articles published in 1995 or later. Even though GDSPs is not a new phenomenon, it was only with the development of communication and collaboration technology during the 1990s that distributed development was made feasible for entire projects [116]. Based on this, we initially considered GDSP research prior to 1995 to be of lesser interest. In the second step, the resulting set of articles was limited to include the 500 most relevant according to the Web of Science analysis tool [107]. This set of articles was further restricted to include only those published in rated journals (see Appendix A). The list of rated journals was a result of a thorough examination of studies of journals in our two areas of research: management information systems [45], [62], [92], [115] and management [29], [33], [43]. The resulting articles of these first two steps were evaluated in the third step based on a detailed examination of abstracts. Articles of little or peripheral interest were excluded from the set. To ensure that key articles in our area of research were included in the final set, the fourth step went backward through the cited references of all articles included by the third step. Articles referenced more than once were evaluated using the third step, exempting the rated journal list, since we considered referencing an acceptable quality indicator in itself. The final set of articles for the review is listed in Appendix B.

B. Structuring the Review

The first part of the review was identification of risks most threatening to distributed projects (Section III). According to Boehm [7], risk areas consist of a number of related risk factors, which together possess a threat to the project's success. Thus, risk areas represent categories of risk factors, where the joint assessment of risk factors indicates whether the risk area might become a problem for a project. We adopted a systematic method to synthesize risk areas: we found inspiration in the categories of risk areas used in key articles with an overall perspective on GDSPs; used Leavitt's [60] model as suggested by Lyytinen *et al.* [67] to provide clear foci for a distinct set of risk areas; aggregated a complete list of risk factors identified in the literature and categorized them according to the proposed risk areas; and finally, provided questions and criteria to offer precise definitions of each risk factor.

The second part of the review focused on identifying and categorizing resolution techniques that address risks through managerial intervention (Section IV). As we found no independent categorization of resolution techniques in the reviewed GDSP literature, we looked for inspiration in the software risk management literature. McFarlan [75] presents a generic software risk management framework that has proven its worth time and again over the past 25 years. McFarlan [75] uses four categories of resolution techniques centered on basic project management disciplines and with a particular focus on integration. As integration is a major challenge in managing GDSPs, we adapted McFarlan's framework to help structure the available resolution techniques.

McFarlan's [75] categories are internal integration, consisting of techniques to support coordination and communication internally in the project group; external integration, consisting of techniques to support coordination and communication with external stakeholders; formal planning, consisting of techniques to support planning; and finally, formal control, consisting of techniques to ensure that the formal planning stays on track and is continuously updated in relation to project practices. The literature on GDSPs is less concerned with the challenges of internal

TABLE I
CATEGORIES OF RESOLUTION TECHNIQUES IN GDSP ADAPTED FROM MCFARLAN [75]

Category	Definition
Planning	Techniques that help plan projects to be effectively executed in distributed contexts.
Control	Techniques that facilitate tracking progress and help manage discrepancies in relation to plans in distributed contexts.
Social integration	Techniques that integrate participants and help manage cultural differences across sites in distributed contexts.
Technical integration	Techniques that increase connectivity and technical compatibility across sites in distributed contexts.

and external integration. Instead, there is considerable focus on how communication and collaboration efforts can be supported by various forms of information and communication technology. Also, social integration is generally considered a key challenge because the presence of several cultures in GDSPs creates an environment significantly different from that of collocated projects. Furthermore, resent research has pointed out control as not only being formal but also informal in GDSPs [15], [54]. On that background, we chose to adapt McFarlan's [75] concepts to the following resolution technique categories: planning, control, social integration, and technical integration. Table I provides definitions of these categories.

III. CONCEPTUALIZING RISKS

In the following, we synthesize risk areas across key GDSP articles supported by Leavitt's organizational model [60], [67]. Subsequently, we characterize each risk area and the risk factors it consists of with references to the reviewed articles.

A. Synthesizing Risks

Leavitt's [60] organization model was developed to synthesize the primary dimensions and dynamics of organizations. According to Lyytinen *et al.* [67], it applies well to define risk in software development into distinct areas: task covers the results, products, approaches, and goals of the software project; structure represents the project organization and institutional setting; actors consist of users, managers, developers, and other key stakeholders; and finally, technology consists of development methods and tools and of the hardware and software platforms for the resulting software.

Based on its merits in defining the foci of different risk areas in software development, we used Leavitt's model to propose distinct risk areas based on key GDSP articles. In addition, we aggregated a complete list of risk factors from the reviewed literature and categorized them according to the proposed risk areas. Table II presents the resulting synthesis of risk areas and related risk factors. The first five columns describe the risk categories found in other key articles. These are related to the proposed risk areas in the second last column. A gray cell denotes that the article does not cover that proposed area. The last column defines the focus of each proposed risk area in relation to Leavitt's four dimensions.

Table II documents in this way: (1) how the proposed conceptualization of risk areas synthesizes key articles in the GDSP literature; (2) how the proposed risk areas represent a balanced view and have distinct foci following Leavitt [60]; and (3) how the complete list of risk factors aggregated from the literature further define each risk area. Elaborate definitions of the questions and criteria needed to assess each risk factor are provided in Table III. These definitions and the foci of risk areas in Table II summarize how the literature has been synthesized into distinct risk areas and related risk factors. The following subsections characterize each risk area and the risk factors it consists of, with references to the reviewed articles.

B. Task Distribution

As in traditional software development, the task represents a possible risk in GDSPs, but for slightly different reasons. When the overall project task is divided and distributed across several sites, task uncertainty emerges, because participants may lack information about the task, its purpose [52], [99], and their own contribution to the overall task [24], [28], [36]. Task uncertainty represents lack of information needed to develop the software [31], [71], [76], and it can result in slow change coordination [36] and process and relational conflicts [99]. Task equivocality, in contrast, represents how well participants understand the specification of the task. For GDSPs, in particular, it is important whether the task is routine or nonroutine and how it relates to the experiences of the project team. High equivocality increases coordination and communication needs [6], [116] and demands on interaction media [99]. Finally, as the task is always distributed in GDSPs, high task coupling between task segments increases the need for intersite communication, coordination, and integration, and it can lead to lower level of performance as well as increase the number of failures [12], [24], [35], [37], [99], [100].

C. Knowledge Management

Knowledge management refers to how projects create, capture, and integrate knowledge about the project task, including goals, problems, possible solutions, and approaches. When GDSP participants lack face-to-face interaction [5], [68], knowledge creation is limited within the organization [68]. This may lead to problems in creating collaboration know-how [17], [68] and domain knowledge [5]. Also, knowledge capture may be limited in GDSPs due to factors such as changing relations and roles across the organization [10], properties of electronic communication media [101], and lacking knowledge of different

TABLE II
SYNTHESIS OF RISK AREAS

Sakthivel [99]	Powell et al. [91]	Townsend et al. [108]	Herbsleb and Moitra [37]	Bell and Kozlowski [6]	Proposed risk areas	Focus [60, 67]
-Task coupling -System Characteristics	-Task-Technology-Structure Fit	-Changing Variety of Assignments and tasks	-Strategic Issues	-Team Task Complexity and Workflow	**Task Distribution** - Task Uncertainty - Task Equivocality - Task Coupling	
	-Training		-Knowledge Management	-Information, Data, and Personal Communication	**Knowledge Management** - Knowledge Creation - Knowledge Capture - Knowledge Integration	**Task**
-Place and Type of Offshore Facility	-Design -Coordination	-Different Geographic Locations		-Spatial Distance -Time Zone Distribution	**Geographical Distribution** - Spatial Distribution - Temporal Distribution - Goal Distribution	
-Systems Development Processes -Management of Virtual Projects	-Training -Coordination	-Structural Resistance	-Strategic Issues -Project and Process Management Issues	-Team Members' Roles	**Collaboration Structure** - Collaboration Capability - Coordination Mechanisms - Process Alignment	**Structure**
-Cultural differences	-Culture	-Cultural Differences	-Cultural Issues	-Spatial Distance	**Cultural Distribution** - Language Barriers - Work Culture - Cultural Bias	**Actor**
-Group cohesion	-Trust -Relationship Building -Cohesion	-Trust and Cohesion Issues			**Stakeholder Relations** - Stakeholder Commitment - Mutual Trust - Relationship Building	
-Interaction Media	-Communication		-Inadequate Communication	-Information, Data, and Personal Communication	**Communication Infrastructure** - Personal Communication - Interaction Media - Teleconference Management	**Technology**
-Interaction Media	-Technical	-Tele- and Information Technologies -Technophobia	-Technical Issues	-Information, Data, and Personal Communication	**Technology Setup** - Network Capability - Tool Compatibility - Configuration Management	

sites [40]. This results in reduced capability to discover defects in the developed software [40] or loss of knowledge about options or specific problem solutions [10]. Knowledge capture is especially important when dissolving the project since it may be difficult to subsequently locate a person who possesses the needed knowledge. Moreover, in GDSPs, changing or unclear organizational structures may lead to limited knowledge integration and sharing [1], [10], [37]. Knowledge sharing may be limited across sites due to noncoherent political agendas, and it may complicate prioritizing assignments appropriately or reduce reuse in software development [37].

D. Geographical Distribution

Distribution of activities in a GDSP occurs along three dimensions: space, time, and goals. Spatial distribution complicates the project manager's ability to monitor participants and progress, increases travel budgets, limits face-to-face interaction, and weakens social relations [5], [6], [20], [25], [100]. Temporal distribution increases the complexity of planning and coordination activities, makes multisite virtual meetings hard to plan [12], [16], [25], [47], causes unproductive waits, delays feedback, and complicates simple things like time referencing and time settings [100]. Besides differences in space and time, goal distribution can potentially lead to conflicts related to task interpretation, process principles, and problem resolution approaches [38] and result in site wars and low performance [38], [77], [86]. Goal distribution is more likely in GDSPs because of faulty transfer of information [38] and focus on own site performance.

E. Collaboration Structure

Collaboration is a relatively broad area that covers risks arising when collaboration structures do not fit the distributed context. Collaboration capability describes the project participants' understanding and appreciation of differences in competencies [10], [30] and their ability to effectively use technology to gather and share information across geographical and functional distances [20], [91]. This is often problematic in GDSPs as participants have limited understanding of other project participants' competencies [91], [97], [100]. GDSPs are often characterized by more horizontal organizational structures [10], and flexibility concerning roles and assignments is, therefore, an important quality [108]. Poor fit between project participants and project organization can lead to conflicts, communication problems, and unused potential [103], [104]. Additionally, it may be difficult to establish effective coordination mechanisms in GDSPs, overcoming challenges such as lacking face-to-face interaction [100], problematic task coupling [12], [94], [99], different time zones, local holidays [100], weak social networks [36], and unclear lines of communication [35], [74]. Problems can be exacerbated by weak alignment of coordination mechanisms between sites or by uncritically transferring

TABLE III
DEFINITION OF RISK FACTORS

Risk Area	Risk Factor	Risk Question	Low Risk	Medium Risk	High Risk
Task Distribution	Task Uncertainty	Do participant posses the knowledge and capabilities needed?	Participants know the task and it fits well with their capabilities.	Participants have reasonable task knowledge and their capabilities cover most challenges.	There are serious gaps in the participants' task knowledge and required capabilities.
	Task Equivocality	Do participants understand the specification of the task?	The task is well specified and understood by participants.	Most aspects of the specification are clear and the task is understood by key participants.	The specification lacks clarity on major points and many participants have limited task understanding.
	Task Coupling	Is the task divided into distinct sub-tasks across sites?	There is minor need to coordinate development work across sites.	There is some need to coordinate development work across sites.	There is major need to coordinate development work across sites.
Knowledge Management	Knowledge Creation	How is task knowledge created across sites?	All sites contribute well to creation of required task knowledge.	Most sites contribute reasonably well to creation of required task knowledge.	There are major problems related to creation of required task knowledge.
	Knowledge Capture	How is task knowledge captured across sites?	Task knowledge is captured effectively across sites.	Task knowledge is with some exceptions captured effectively across sites.	There are major problems related to capturing task knowledge across sites.
	Knowledge Integration	How is task knowledge integrated and shared across sites?	Task knowledge is integrated and shared well across sites.	Task knowledge is with some exceptions well integrated and shared across sites.	Task knowledge integration and sharing across sites is limited.
Geographical Distribution	Spatial Distribution	How many sites are involved and what is the distance between them?	There are few sites collaborating over limited distance.	There are several sites collaborating over some distance.	There are many sites collaborating over considerable distance.
	Temporal Distribution	How do time-zone differences impact development work?	Time-zone differences cause no or only minor problems.	Time-zone differences require some ad-hoc coordination across sites.	Time-zone differences cause major problems and require constant attention across sites.
	Goal Distribution	How diverse are goals across sites?	Participants share major goals across sites.	There is some variation in goals across sites.	There are major goal conflicts across sites.
Collaboration Structure	Collaboration Capability	Can participants collaborate across sites?	Participants collaborate across sites as needed.	In most cases, participants collaborate across sites as needed.	Breakdowns in collaboration across sites are common.
	Coordination Mechanisms	Are coordination mechanisms appropriate across sites?	Coordination mechanisms are shared across sites and well adapted to the distributed context.	Coordination mechanisms are shared by most participants and reasonably well adapted to the distributed context.	Coordination mechanisms are not shared across sites and poorly adapted to the distributed context.
	Process Alignment	Are software processes aligned across sites?	Software processes (including methods, templates, and guidelines) are shared across sites.	Software processes (including methods, templates, and guidelines) vary, but are reasonably well aligned across sites.	Software processes (including methods, templates, and guidelines) are different across sites.
Cultural Distribution	Language Barriers	Do language and communication norms vary across sites?	Participants share language and communication norms across sites.	Participants use a common language with minor differences in communication norms.	Participants do not share a common language and have different communication norms.
	Work Culture	Does work culture differ between sites?	Participants share work culture (including authority and team behavior) across sites.	Participants understand variations in work culture (including authority and team behavior) across sites.	Participants do not understand variations in work culture (including authority and team behavior) across sites.
	Cultural Bias	Does cultural bias impact communication and cooperation across sites?	There are no major variations in cultural values across sites.	Participants communicate and collaborate based on appreciation of cultural variations across sites.	Participants lack knowledge of variations in cultural values across sites.

TABLE III
CONTINUED

Stakeholder Relations	Stakeholder Commitment	Are stakeholders committed to the project?	Key stakeholders are committed and share a common project identity across sites.	Most stakeholders are committed to the project and know about its distributed organization.	Stakeholder commitment varies and there is a lack of shared project identity.
	Mutual Trust	Is there trust between stakeholders across sites?	There is appropriate mutual trust across sites.	There are instances of insufficient trust across sites.	Stakeholders do not trust each other across sites.
	Relationship Building	Can the project integrate stakeholders across sites?	Existing and new stakeholders are well integrated across sites.	Existing and new stakeholders are mostly integrated well across sites.	There are several cases of stakeholders not being well integrated.
Communication Infrastructure	Personal Communication	What is the level of personal communication and social interaction across sites?	The level of personal communication and social interaction across sites is appropriate.	There is some personal communication and social interaction across sites.	Personal communication and social interaction across sites is limited.
	Interaction Media	How is communication across sites supported by media?	Communication needs across sites are well supported by media.	Communication across sites is for many purposes well supported by media.	There are severe shortcomings in media support of communication across sites.
	Teleconference Management	How is teleconferencing managed across sites?	Teleconferencing is used appropriately and managed effectively across sites.	Teleconferencing is used to some extend across sites and reasonably well managed.	There is limited use of teleconferencing across sites and they are not managed well.
Technology Setup	Network Capability	Are communication networks reliable?	Networks are not causing delays in development work and communication.	Networks are causing some delays in development work and communication.	Networks are causing serious delays in development work and communication.
	Tool Compatibility	Are tools compatible across sites?	There are no compatibility issues between tools across sites.	Compatibility issues between tools create some collaboration barriers across sites.	Compatibility issues between tools create serious collaboration barriers across sites.
	Configuration Management	How are configurations managed across sites?	There is appropriate configuration management across sites.	Configuration management is with some exceptions appropriate across sites.	There is limited configuration management across sites.

nonapplicable coordination mechanisms from collocated projects to GDSPs [91]. Process alignment in terms of traditions, development methods, and emphasis on user involvement will often differentiate between sites, possibly resulting in incompatibility and conflicts [5], [24], [25], [56], [99], [100].

F. Cultural Distribution

When projects are geographically distributed, a number of cultural problems may arise since participants do not necessarily share the same language, traditions, or organizational culture [80], [93], [106]. Language barriers arise in cross national projects when sites and participants do not share a common language [12], [16], [22], [24], [37], [56], [88], [91], [93], [100] or norms of communication [22], [24], [37], [47], [91], [100], [108] resulting in misinterpretations and unconveyed information [22], [88], [100]. Differences in work culture may render difficulties in a GDSP [22] when sites are different in terms of team behavior [22], balancing of collectivism and individualism, perception of authority and hierarchy [37], [56], [88], planning, punctuality [37], and organizational culture [12], [16]. This may lead to decreased conflict handling capabilities and lower efficiency [78], [85], or even paralyze the GDSP [22]. Cultural bias occurs when project participants consider their norms and values as universal and neglect to reflect on to what extent values, norms, and biases are founded in their own cultural background [22], [104]. Cultural bias may lead to erroneous decisions [22] and insecurity about other participants' qualifications [5], and it can have a devastating impact on communication and collaboration efforts [47], [88].

G. Stakeholder Relations

When projects are distributed, it naturally becomes difficult to obtain the same level of stakeholder integration as you would expect in a collocated organization [99]. Lack of frequent face-to-face interaction may impair relationship building [22], [87], [91], [99] since relations are build through communication between project stakeholders [87]. The problem also extends to integration of new project participants [6] and other stakeholders in the organization [84]. Closely related to stakeholder relations is the question of trust. Mutual trust is important but hard to obtain in GDSPs [58], [61], [98], [106]. This can be due to lack of face-to-face interaction [20], [74], [109], cultural differences, and weak social relations [87]. Trust among stakeholders is necessary to achieve innovation, flexibility, cooperation, and efficiency in a distributed environment [2], [13], [20], [30], [42], [44], [59]. Furthermore, since GDSPs often have a short life span, it is important to achieve mutual trust rapidly [41], [91], but if trust is misplaced, the entire organization may suffer [59]. Ultimately, relationship building and mutual trust problems extend to lack of stakeholder commitment [61]. Stakeholders are less likely to commit to the project organization and its task when cultural differences and lack of face-to-face interaction makes it difficult to establish a clear project identity [6], [30], [78]. This weakens group synergy [6], [30], increases the risk of conflicts [78], and may lower efficiency in the initial project phase [105].

H. Communication Infrastructure

Almost every problem arising in GDSPs is related to the fact that communication is no longer a simple task when participants are distributed and appropriate supporting infrastructures are therefore needed. Personal communication is often impeded by absence of informal communication [35], [37] and lack of face-to-face interaction [22], [87]. This can negatively impact trust [47], [100], decision quality [37], [47], creativity [47], [100], and general management [20]. Furthermore it may reduce participants' project overview, which can lead to errors and misunderstandings [37]. Being separated, interaction media becomes the primary communication link between sites, but their properties or use may cause problems such as jumbled sequences of messages; mix-ups between past, present, and future messages [47], [70], [100], [112]; and loss of contextual information sharing [47], [111]. Such problems, arising with either synchronous or asynchronous interaction media, may lead to confusion [100] and misunderstandings [111] among participants and lower the moral [47]. GDSPs are highly dependent on proper teleconference management in order to coordinate efforts between sites. When interaction medium limits verbal and nonverbal cues, it is not possible to apply traditional management of meetings [100], [112]. Additionally, different time zones may make it difficult to organize conferences [5], [12], [88]. These factors make it challenging to benefit from conferences [13], [112].

I. Technology Setup

Networks that connect globally distributed sites are often slow and unstable [22], [37], and even minor delays can ruin the flow of communication [17], [46], [47], [70], [88], [100], [112]. Network capability is therefore an important challenge in GDSPs, and selection of appropriate information and communication technology is crucial for project success [46], [68], [110]. Unreliable networks may lead to frustration and low efficiency [22], limit exchange of sensitive information [10], [88], or even cause production stop [22]. When developers from different parts of the world collaborate, tool compatibility may prove a problem. The reason is that sites are likely to prefer different programming languages, support tools, operating systems, and development tools [22], [46], [100]. Also, the sites may experience differences in support and tool versions. This can lead to frustrations, conflicts, and delays [100]. Configuration management is specifically a challenging technology in distributed projects due to possible problems concerning tool differences [5], slow and unreliable sites, lacking awareness of product changes, and bug fixes between sites [24], [37], [40].

IV. Conceptualizing Risk Resolution Techniques

Thirty-five risk resolution techniques were identified in the reviewed literature. In the following, we present these techniques using the four categories: planning, control, social integration, and technical integration (cf. Table I). The categories were, as mentioned earlier, adapted from McFarlan's [75] generic software risk management framework. We have categorized each resolution technique based on its primary emphasis as each technique very well can apply to more than one domain. Due to the amount of resolution techniques, we opted not to elaborate each in detail; instead, we present one exemplary resolution technique for each category of resolution techniques: a complete list of the identified resolution techniques is presented in Appendix C.

A. Planning

The planning category includes resolution techniques that help plan projects to be effectively executed in distributed contexts. An important planning technique in GDSPs is "create shared collaboration platform," offering a shared vocabulary to describe both everyday activities on each site and central activities in the development process, e.g., by using UML. This promotes unity and sense of belonging and reduces misinterpretations [5], [16], [100]. Also, it is advised to establish a shared project culture without discriminating in favor of any particular national or professional culture. More specific suggestions are the use of a dialogue technique to establish shared mental models of the project and task [105] and the production of concept lists explaining slang across the involved cultures and professions [69].

B. Control

The control category includes resolution techniques that facilitate tracking progress and help manage discrepancies in relation to plans in distributed contexts. An example of a technique that supports control in GDSPs is "establish temporal coordination mechanisms," providing structured approaches to temporal coordination across sites including handling of deadlines, synchronization, and distribution of resources [70], [77]. Shared deadlines or milestones should be introduced when coordinating successive integration of individual software modules as well as handling diversities concerning local festivals and holidays [5], [100]. If reduction of temporal distance is impossible, the project manager should manage time translations and time adjustments, relocate time using asynchronous media, and institute time-based norms for communication and virtual presence [100]. There should also be a focus on synchronization, plans, and procedures in the development process, enabling transferring of tasks from one site to another [99].

C. Social Integration

The social integration category covers resolution techniques that integrate participants and help manage cultural differences across sites in distributed contexts. One of these techniques is "develop liaisons between sites." This approach advocates using liaisons to facilitate information exchange, identify expertise, mediate cultural conflicts, and settle disputes [5], [12], [24], [35], [36], [56]. If the organization consists of a main site and several subsites, liaisons from the subsites should spend the start-up phase at the main site to gain insight and overview of the project [5]. Furthermore, it is advised to include travel expenses in the overall budget and not perceive them as additional costs.

D. Technical Integration

The technical integration category includes resolution techniques that increase connectivity and technical compatibility across sites in distributed contexts. An example of these resolution techniques is "standardize and train in methods across sites." This technique suggests standardization of tools, methods, templates, and processes in order to create a harmonic and efficient project organization [24], [25], [56], [99]. Such standardization implies training of participants and lower initial efficiency as experience with the chosen standards varies [5], [25], [46]. In the long run, higher efficiency and fewer misunderstandings are, however, expected [24], [56]. More specific standardization could be introduction of shared guidelines for error handling, accessibility to other sites' documentation, documentation of tests and testability [40], or the use of a shared tool that allows for tracking of bugs and corrections in all parts of the distributed project [5].

V. An Integrative Framework

Based on the insights from the two previous sections, we developed a framework for managing risks inherent in GDSPs. First, we linked resolution techniques to risks based on the literature. Then, we developed a framework for risk assessment, risk control, and risk management planning in GDSPs. Finally, we implemented a Web-based tool called Distributed Project Management System (DPMS) to help refine the framework based on empirical evaluation of its practical usefulness. In the following, we present each of these steps in detail.

A. Linking Resolution Techniques to Risks

Based on the reviewed literature, we linked resolution techniques to risk areas, reflecting which resolution techniques can alleviate which risk areas. To that end, we created a matrix with risk areas on the x-axis and resolution techniques on the y-axis (Table IV). The reviewed articles were then revisited and references were added to the matrix where we identified a link in the literature. This identification process was interpretive rather than literal.

B. Developing Framework

Subsequently, we considered four classical risk management frameworks presented by Lyytinen *et al*. [67]. These were McFarlan's [75] portfolio framework, Davis's [18] contingency framework, Boehm's [7] software risk framework, and Ginzberg's [32] implementation framework. The design of Ginzberg's approach [32] did not qualify for our purpose because it does not include risk resolutions. Of the remaining three, we opted for the design of Boehm's [7] risk-action list framework, as it possesses two important qualities: it is easy to use and modify [39]. We considered ease of use crucial, as the framework should be employed in a distributed and likely cross-cultural context. In such a setting, we made it a priority to enable participants with varying background to use the framework without lengthy preparatory instructions. Regarding easy modification, flexibility is a desirable trait in GDSPs as these organizations have changing needs [22], [37], [108]; also, we considered future development of the framework, taking into account that rapid development of technology and organizational forms plays a major role in GDSPs, making future alterations inevitable. The tradeoff when comparing risk-action list frameworks (e.g., Boehm [7]) with risk-strategy frameworks (e.g., McFarlan [75]) and risk-strategy-analysis frameworks (e.g., Davis [18]) is lack of strategic oversight [39]. However, we did not consider this as important as the other qualities because the proposed risk management framework focuses on risks related to geographical distribution rather than on risks in general. The framework therefore demands complementary management and risk management approaches to provide appropriate strategic oversight.

According to Boehm [7], risk management involves risk identification, risk analysis, risk prioritization, risk management planning, risk resolution, and risk monitoring. Our risk management framework consists of three elements, formalizing these steps in GDSPs (Fig. 2). In terms of content, the framework differs from Boehm's [7] by specifically focusing on risks related to distributed projects; we exclude risks appearing in collocated projects unless the distributed environment significantly exacerbates them. Additionally, during risk assessment, our framework estimates risk exposure on the risk factor level, opposed to Boehm [7], who evaluates risk exposure on the risk area level.

Our framework supports multiple users in order to engage project participants across sites as illustrated in Fig. 2. This is a crucial feature for risk assessment accuracy, as no single project manager possesses the necessary overview of GDSPs to accurately perform risk management. Another important structural feature is the support of project hierarchies in GDSPs [26]. This allows for subprojects within distributed projects to contribute to an overall risk assessment.

When applying our framework to a GDSP, the first step is risk assessment. This process evolves around a model with the eight identified risk areas and 24 risk factors that constitute the results from the literature review (Fig. 3). For each risk factor, the user selects a risk probability P(UO) (defined as the probability of unsatisfactory outcome [7]) and the loss to the parties affected if the outcome is unsatisfactory L(UO) [7]. These assessments are made on a scale with the numeric values 0–8, categorized into low (0–2), medium (3–5), and high (6–8) (Fig. 3). During probability assessments, users are supported by not only the numeric scale but also by a qualitative interpretation of each measure based on the literature review; see example in Fig. 3 or Table III for the full list. After users have assessed both

TABLE IV
LINKING RESOLUTION TECHNIQUES TO RISK AREAS

Planning	**Task Distribution**	**Knowledge Management**	**Geographical Distribution**	**Collaboration Structure**	**Cultural Distribution**	**Stakeholder Relations**	**Communication Infrastructure**	**Technology Setup**
Acquire complementary skills	[10, 103]		[10, 20, 103, 108]	[20, 103, 108]	[69]	[78, 103]	[112]	[10, 103]
Adjust meetings to distributed context	[88]	[13, 22, 56, 112]	[13, 88, 100, 112]				[13, 22, 100, 112]	[13]
Divide tasks systematically between sites	[5, 24, 35, 36]	[5]		[35, 36]				
Reduce coupling between sites	[91, 94]		[12, 91, 94]	[12, 91, 94]				
Create shared collaboration platform		[5, 16, 100, 105]	[5, 16, 77, 100]	[5, 100, 105]	[16, 24]	[68]		[5]
Establish shared goals	[10, 52, 108]	[46]	[6, 10, 30, 46, 52, 104, 108]	[108]		[6, 46, 58, 98, 104]		
Establish communication norms		[47, 88, 100, 111, 112]	[13, 88, 100, 117]		[24]	[69, 111]	[13, 47, 88, 100, 108, 112, 117]	[47, 88, 100, 111, 112]
Define roles and responsibilities			[59, 100]	[117]	[24, 41, 59, 100]	[59]		
Reduce time-zone differences			[12, 99]					
Control	**Task Distribution**	**Knowledge Management**	**Geographical Distribution**	**Collaboration Structure**	**Cultural Distribution**	**Stakeholder Relations**	**Communication Infrastructure**	**Technology Setup**
Focus on deliverables		[37]	[77]					[37, 46]
Establish task coordination between sites	[5, 24, 37]			[24]				
Maintain site autonomy	[5]		[5, 6, 25, 52, 89]	[25]		[89]	[5, 52]	
Establish shared control mechanisms	[6]	[25]	[6, 20, 25, 44, 104, 110]	[25]		[20, 44]		
Establish temporal coordination mechanisms	[5]	[99]	[5, 70, 77, 99, 100]	[99]				[100]
Maintain project organization overview	[28]	[28, 35, 36, 100]	[36, 100]	[35]			[74]	[28, 36, 100]
Maintain task overview within and across sites	[5, 6, 28]	[28, 68]	[68]				[68, 105]	[28, 68]
Monitor and improve communication		[88]	[110]			[44]		[88]
Maintain a supportive environment	[82, 97, 110]		[59, 85, 100]		[59]	[58, 59, 85]		[97]
Analyze and manage errors	[40]	[40]						
Social Integration	**Task Distribution**	**Knowledge Management**	**Geographical Distribution**	**Collaboration Structure**	**Cultural Distribution**	**Stakeholder Relations**	**Communication Infrastructure**	**Technology Setup**
Improve capability to manage cultural differences		[22]	[22, 104]		[22, 56, 100, 104]			

TABLE IV
CONTINUED

	Task Distribution	Knowledge Management	Geographical Distribution	Collaboration Structure	Cultural Distribution	Stakeholder Relations	Communication Infrastructure	Technology Setup
Improve distributed collaboration skills			[13, 30, 85, 91, 100, 104, 108, 111, 117]	[13, 91, 104]		[30, 85]	[13, 108, 112]	
Improve language skills					[12, 22, 24, 88]			[22]
Emphasize early teambuilding activities			[24, 46, 77, 97, 110]			[24, 46, 58, 97, 110]	[35]	[97, 110]
Promote humor and openness			[47, 91]		[91, 104]	[47]		
Use mentors to integrate new members	[6, 30, 105]	[6, 30, 105]				[6, 30, 105]	[105]	
Use face-to-face meetings appropriately	[73, 99]	[10, 97, 110]	[30, 35, 64, 73, 87, 91, 97, 99, 110, 112, 117]	[10]		[30, 35, 64, 87, 91, 97, 99, 110]	[52, 73, 112, 117]	
Develop liaisons between sites	[5, 12, 24, 35, 36, 56]	[5, 12, 24, 35, 36, 56]	[5, 12, 24, 35, 36, 56]		[5, 12, 34, 56]	[35]	[5, 35, 36]	
Adopt shared reward systems				[10]	[108]	[30, 64]		
Technical Integration	**Task Distribution**	**Knowledge Management**	**Geographical Distribution**	**Collaboration Structure**	**Cultural Distribution**	**Stakeholder Relations**	**Communication Infrastructure**	**Technology Setup**
Increase technical compatibility between sites		[10]	[10, 22, 46]				[10]	[5]
Standardize and train in methods across sites	[40]	[5, 40]		[5, 25, 99]	[24]			[40, 46]
Adopt appropriate communication technologies	[19]	[1, 10, 17, 101]	[20, 30, 44, 47, 64, 85, 97, 100, 110, 111]		[56, 88]	[30, 44, 47, 64, 90, 111]	[10, 35, 36, 90, 110]	[10, 22, 68-70]
Improve collaboration and communication technology skills		[5, 10, 30, 46, 68, 91, 108]	[68]	[68, 91]		[111]	[5, 10]	[22, 30]
Improve development technology skills		[22, 46, 91, 100]						[91]
Handle differences in methods between sites		[100]		[5, 25, 100]				
Combine waterfall model and prototyping	[99]			[99]				

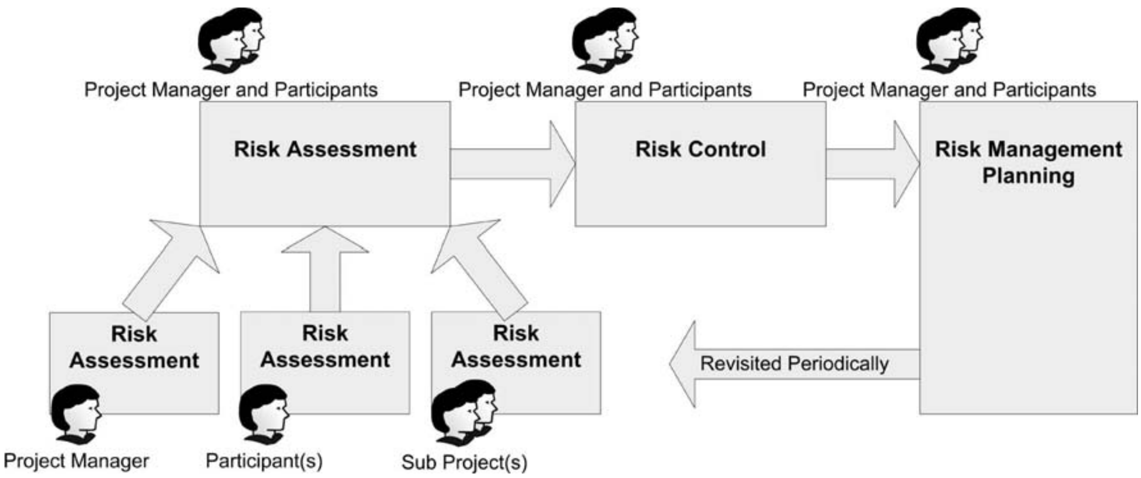

Fig. 2. Framework elements.

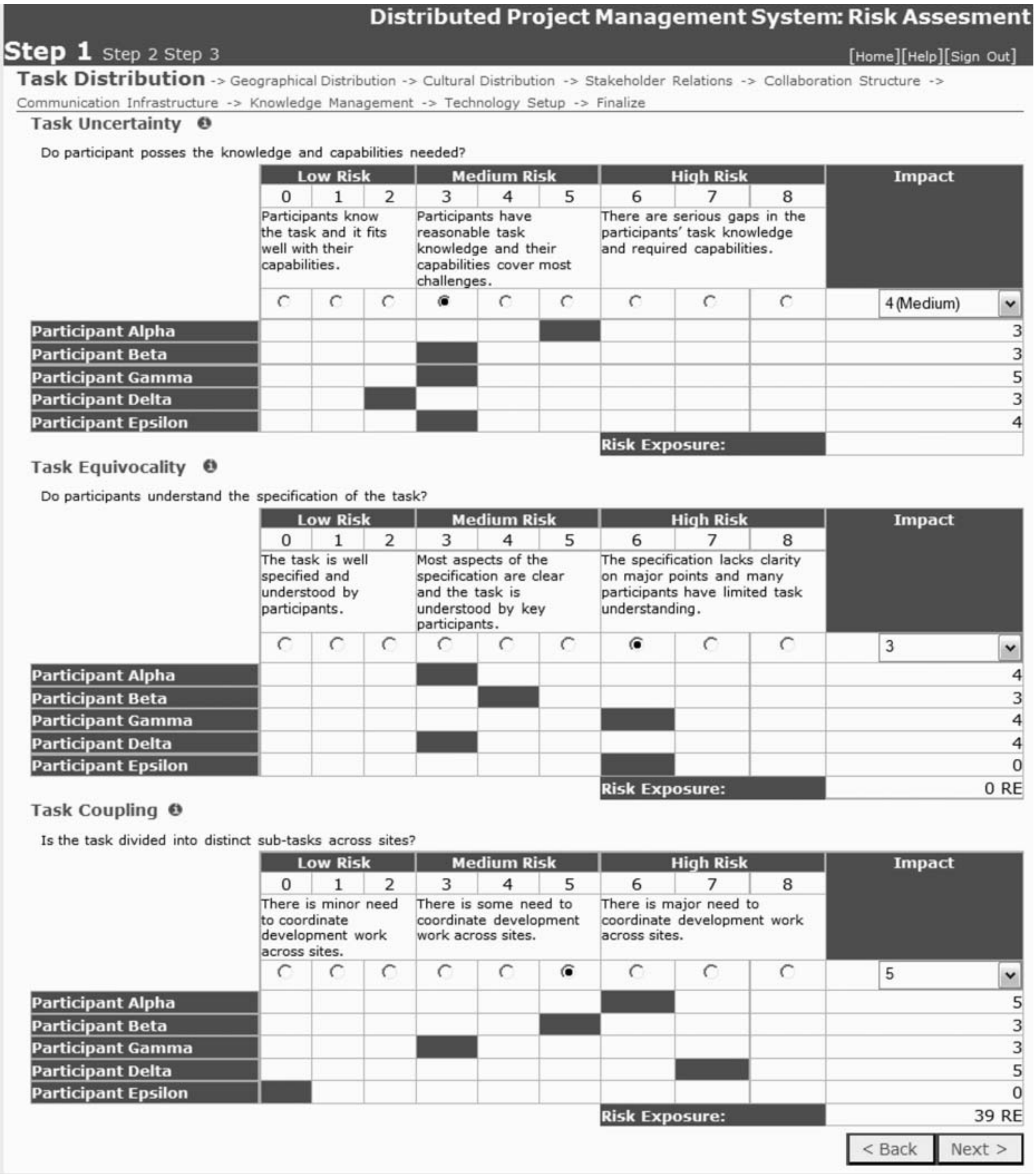

Fig. 3. Risk assessment in DPMS.

the risk probability and impact for a given risk factor, the risk exposure (RE) is calculated based on the equation RE $= P$(UO) $\times L$(UO) proposed by Boehm [7]. The average RE for the three risk factors subsequently constitutes the risk area's RE value. Based on these values, a prioritized list of the eight risk areas is derived, representing the significant risk areas in the GDSP.

A number of GDSP participants independently perform a risk assessment. The combined set of assessments subsequently forms the basis for a risk discussion among the participants where the participants share the submitted assessments, as shown in Fig. 3, to allow for direct comparisons. The participants, either collocated or mediated via conference, can then negotiate a shared risk assessment for the GDSP. In this process, the project manager and participants can obtain valuable insights into and overview over the GDSP [39], [67]. It is the responsibility of the project manager to keep the discussion structured and update the risk assessment according to findings. The discussion may be lengthy or short depending on factors such as differences in opinions and discussion management. The ideal result of the discussion is a risk area prioritization that all participants agree upon.

The risk control step following the assessment also relies upon discussion and knowledge sharing among project participants. The first task is to prioritize which risk areas to address using the risk areas' RE values as support. Next, a set of appropriate resolution techniques is presented for each prioritized risk area based on the heuristics matrix (cf. Table IV), represented by checkmarks linking risk areas to resolution techniques in Fig. 4. The project participants then discuss and choose the resolution techniques considered appropriate for each risk area.

The assessments and selected actions are summarized in risk management plans for each of the prioritized risk areas. These plans lay out the activities necessary to bring the related risk area under control. Each plan contains answers to five basic questions, as proposed by Boehm [7]. The objectives (why) are identified through the risk assessment. The deliverables and milestones (what and when) suggest when the selected actions are to be taken. The project manager is free to structure this as best fits the project. The third area, responsibilities (who and where), describes which individuals are responsible for a given task, and where within the distributed organization, it is to be carried out. The approach (how) consists of the previously identified resolution techniques. The fifth and final area is resources (how much), where the participants estimate the costs associated with addressing the risk area under consideration.

The final step in risk management is to integrate the resulting risk management plans with the overall project plan. This process is not directly supported by the framework due to the diversity of project management methods available. In conclusion, a date should be set for revisiting risk management in order to keep risk management plans up to date with the evolving GDSP.

C. *Implementing Web-Based Tool*

To allow us to refine the framework based on empirical evaluation of its practical usefulness, the framework was subsequently implemented as a Web-based tool. This DPMS tool implements and elaborates the proposed risk management framework to support practical management of GDSPs. The tool is available at http://www.distributedprojects.net. While the framework can also be adopted without this tool or with alternative tools, the presented Web-based tool follows the structure depicted in Fig. 2. As an initial step, the project manager is responsible for registering a project and assigning participants and subprojects. When performing risk assessments, users have the opportunity to draw upon supportive information via hyperlinks. Upon completion, the individual risk assessments are submitted and aggregated to support risk discussion. This is done by arranging the individual risk assessments next to each other on the screen, allowing for direct comparison (Fig. 3). When the shared risk assessment is completed, a prioritized list of risk areas is presented. To visually aid the users in selecting the proper risk areas to proceed with, risk areas are color coded: RE $> 47 =$ red, 9 $<$ RE $< 48 =$ yellow, and RE $<10 =$ green. In the following step, risk control, users are presented with a schema based on Table IV that illustrates how resolution techniques apply to risk areas. The prioritized risk areas are highlighted. Users then, for each of the chosen risk areas, select a number of resolution techniques. In this process, users can access elaborate information about each resolution technique via hyperlinks (Fig. 4). The final step is risk management planning. The system automatically fills in information regarding "why" and "how" based on results from previous steps. As this information is generic, it can be edited to fit the specific context. The risk management plan is stored and can later be retrieved when revisiting the DPMS.

VI. EVALUATION OF FRAMEWORK

We developed the risk management framework iteratively. The practical usefulness of each version was evaluated, and the findings were fed into the next iteration. In total, four evaluations were conducted with increasing focus on practical usage: Evaluation I focused on the initial conceptualization of risks and risk resolutions through a focus group; Evaluation II focused on paper-based risk assessment and risk management through a focus group and a workshop; Evaluation III focused on tool-based risk management through a workshop; and, Evaluation IV focused on full-scale application of the DPMS tool with multiple participants in a real-world setting. All four evaluations were documented through field notes, audio recordings, and work documents. Table V summarizes the evaluations. The iterative development–evaluation process was terminated at this point as Evaluation IV only led to minor changes and all participants found the risk management framework useful and easy to use. While the practical usefulness of the framework in this way was evaluated with experienced GDSP practitioners, its effect in complex GDSP contexts was not thoroughly evaluated. Additional evaluations that include effects over longer periods of time are, therefore, an important direction for future research, as discussed later.

A. *Evaluation I: Focus Group*

Evaluation I was carried out as a focus group interview attended by six practitioners representing four different companies: Alpha (2), Beta (1), Gamma (2), and Delta (1). These were midsized to large companies within the information technology (IT) industry with software development activities in multiple countries. The companies were chosen due to their usage of both internationally and nationally distributed software projects. The respondents were all project managers with relevant experience in GDSP and risk management. Each project manager had a master's degree equivalent education in computer science or management information systems and minimum five years of industry experience. The evaluation was divided into an inductive and a deductive part. The first part was an explorative, semistructured focus group interview that had as objective to obtain data about the practitioners' own experiences in GDSPs.

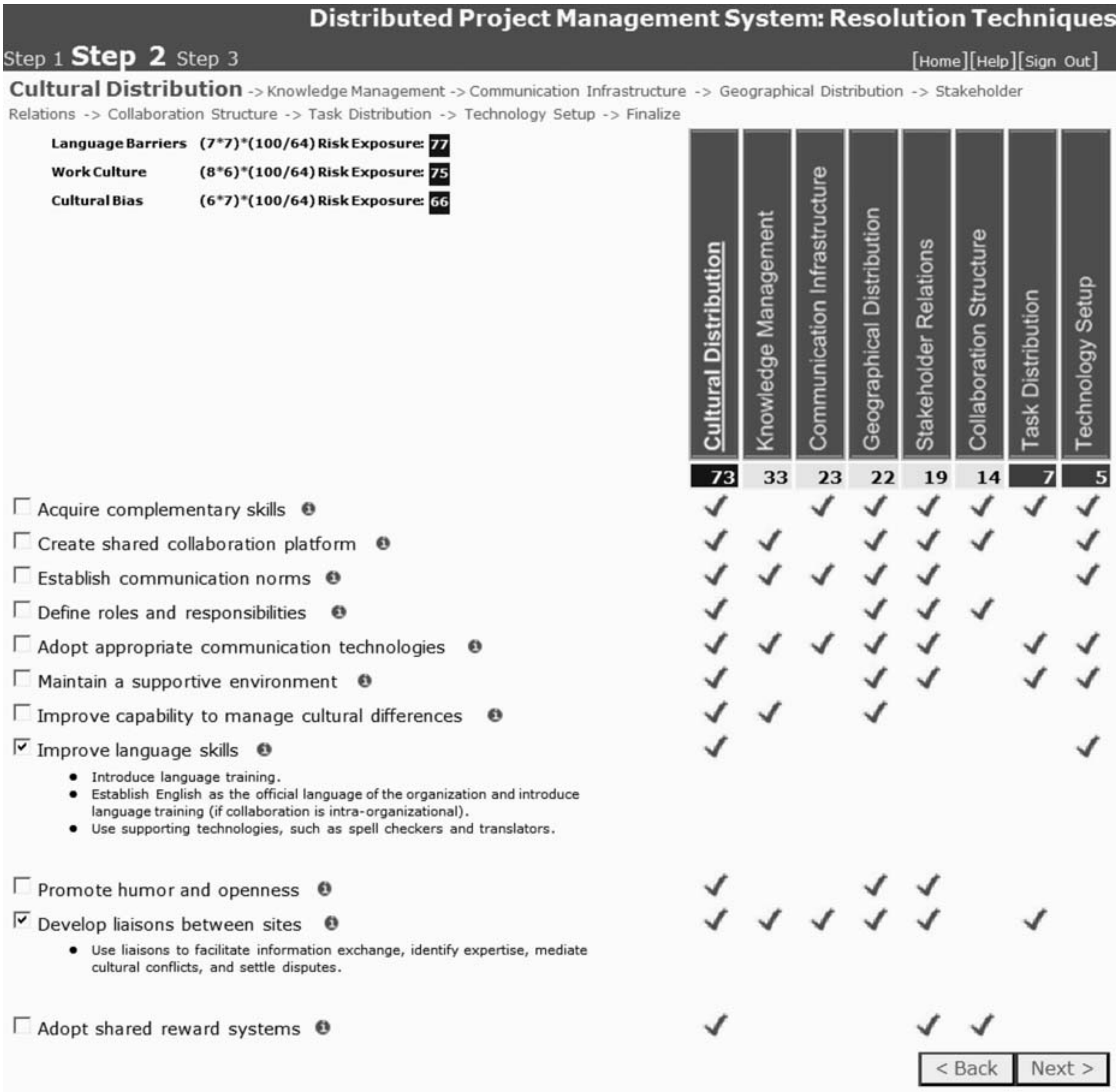

Fig. 4. Resolution technique selection in DPMS.

The second part was a structured focus group interview that evaluated the proposed conceptualization of risks and resolution techniques. During the first step, the practitioners produced two lists: one containing their view of the ten most significant challenges of GDSPs and the other containing the ten best-suited resolution techniques to address the challenges of GDSPs. For both challenges and resolutions, the participants then had to merge the lists into one through discussion and exchange of experiences. In step two, the participants were presented with the result from the first development cycle: a list of risks and a list of resolution techniques. Each entry in the lists was presented in turn, and the participants were asked to rate the importance of the entry according to their own experience as well as evaluate the communicative value of the adopted formulations. In addition, the practitioners were asked to relate each resolution technique to any number of risk areas they thought relevant.

The findings of the first part showed few inconsistencies between the GDSP challenges and resolutions reported by the practitioners and the lists we had derived from the literature. As was to be expected due to the relative small sample size and the exploratory nature of the evaluation, not every entry of the lists was touched upon during the first part. However, the challenges and resolutions presented by the participants helped us rethink

TABLE V
EVALUATION OVERVIEW (RA: RISK AREA; RF: RISK FACTOR; RT: RESOLUTION TECHNIQUE)

	Evaluation Approach	Evaluation Focus	Data Collection	Number of Participants	Resulting Changes
Evaluation I	Focus Group I	List of risks and risk resolutions	- Semi structured interview - Structured interview	6	- Restructuring: 2 RF - Rewordings: 6 RA; 15 RT - Deletions: 3 RF; - Additions: 5 RF; 3 RT
Evaluation II	Focus Group II	Risk assessment (Paper version)	- Prototype use - Debate - Semi structured interview	5	- Restructuring: 8 RF - Rewordings: 25 RF; 1 RA; 21 RT - Deletions: 2 RA; 1 RF - Additions: 2 RA; 4 RF
	Workshop I	Risk management (Paper version)	- Prototype use - Think aloud - Semi structured interview	1	
Evaluation III	Workshop II	Risk management (Electronic version)	- Framework use - Semi structured interview	2	- 8 major usability improvements and 24 RF support text rewordings
Evaluation IV	Workshop III	Risk management (Electronic version and collocated discussion)	- Framework use - Participant observation - Debate	14	- Rewording of 5 RF and 24 RF support texts

the overall risk categorization and the individual risk factors. The comments to and structured evaluations of the framework during the second part also provided valuable input regarding the communicative qualities of both the adopted conceptions and explanatory texts of risk factors and resolution techniques. A number of formulations were therefore improved. The changes in conceptualizations and approach resulting from Evaluation I are summarized in Table V.

B. Evaluation II: Focus Group

The focus of the first part of Evaluation II was to evaluate the practical usefulness of the framework through discussion and use of individual framework elements. To do this, we organized a focus group consisting of five practitioners from Beta: two project managers, a tester, a system architect, and a developer. This sampling represented two nationalities, Danes and Spaniards, as well as five different projects within Beta. The group was initially introduced to the framework, after which each participant performed a risk assessment and risk area prioritization of their respective projects. Following this, the framework and its usability were discussed and its content and structure evaluated. The second major part of the framework, the selection of suitable resolution techniques and elaboration of a risk management plan, was evaluated based on an exemplary walk-through followed by a semistructured interview of the focus group.

The collected data pointed at a number of areas where the framework needed improvement. One of the most significant areas was that all practitioners, to some extent, had difficulties distinguishing between risk areas and risk factors. This led to confusion and, in some cases, to faulty risk analyses. Additionally, some risk areas were seen as being too interdependent. These findings led to major revision of the framework, rephrasing and restructuring both risk areas and risk factors. Suggestions regarding implementation of the framework as a Web solution gave inspiration for the continued development of the framework. Further remarks emphasized using risk assessment as a collective discipline involving a broad selection of project participants as well as considering how the design impacted the general flexibility of the risk management process, e.g., adding the option to alter the calculated RE before proceeding to create a risk management plan. Finally, the evaluation pointed out that the framework did not sufficiently include configuration management, which was seen as a major risk factor in GDSPs. These remarks were carefully considered and subsequently implemented into the framework. The changes in the framework resulting from the focus group part of Evaluation II are summarized in Table V.

C. Evaluation II: Workshop

The second part of Evaluation II was carried out in a workshop setting. The evaluated version of the framework corresponded to the one evaluated in the second focus group interview. There were two objectives: to evaluate the individual parts of the framework (risk assessment, risk control, and risk management planning), the information transfer between them, and the users' understanding of them and to evaluate the overall usability of the framework and its contribution to management of GDSPs. The workshop had only one participant, a project manager at Gamma with extensive experience in both GDSPs and risk management. The participant was introduced to the framework and then proceeded to apply it to his current project. While using the framework, he was asked to think aloud and account for his choices. This approach was inspired by the think aloud test, frequently used in software usability testing [96]. Upon completion of the risk management plan, the participant was debriefed using a semistructured interview.

The overall assessment was that the framework was practically useful; especially, the coupling between risk areas and resolution techniques gave rise to positive feedback. The project manager found that texts in the framework had an appropriate level of abstraction and generally were easy to read. Some risk factors caused confusion, which further contributed to revising risk areas and risk factors (see section VI-B). In general, the participant was reluctant to read the explanatory texts associated with each risk factor. This was taken into account as it further emphasized demands for short, precise formulations. Additionally, comments were given that correspond well to the findings of the focus group part of Evaluation II—configuration management should be emphasized in the model—and it was considered important that project participants participate in the risk assessment as no project manager has sufficient overview to perform it alone. The changes in the framework resulting from the workshop part of Evaluation II are summarized in Table V.

D. Evaluation III: Workshop

The third evaluation took place during the third iteration of developing the framework. At this stage, the risk management framework was implemented as the DPMS tool. The evaluation focus was fourfold: usability, presentation of content, distributed usage, and workflow. To that end, we arranged a workshop with a project group at Alpha that was collaborating with an Indian business partner. Two members of the project participated, the project manager and the business developer. The workshop was divided into two parts, held with one-day interval. During the first part, the two participants completed a risk assessment of their project separately and independently. The project manager performed the initial steps of registering a new project and adding participants to the system before proceeding to make the risk assessment, whereas the business developer performed the risk assessment. Similar to the second part of Evaluation II, both were asked to think aloud, allowing us to follow their train of thought [96]. In the second part, the two participants were brought together and used the tool to perform a shared risk assessment, select appropriate resolution techniques, and create a risk management plan for the project. Upon completion, we debriefed the participants using a semistructured interview.

The participants found the risk assessment part of the framework useful and easy to use, both during the individual assessment and the combined assessment. However, the initial registration process needed improvement as the project manager found it somewhat confusing. The coupling between risk areas and resolution techniques also caused initial problems as the tool did not clearly indicate the process status. Presentation-wise, the risk management plan proved problematic as the elaborating text provided by the tool was perceived as inhibitory rather than helpful. The participants requested greater flexibility to customize the content to their particular project. Concerning distributed usage, the shared risk assessment and the related discussions were considered beneficiary and well supported by the tool. These findings led to revision of the visual presentation of the framework taking the identified usability problems into consideration. Additionally, the risk management plan was altered as the elaborate texts were made accessible to the user during each step rather than presented upfront. Finally, indicators of progress were added to each step of the framework to visualize the user's status in the process. The changes in the framework resulting from Evaluation III are summarized in Table V.

E. Evaluation IV: Workshop

Evaluation IV was carried out in a project at a large fifth company, Epsilon, not involved in the previous evaluations. The evaluation focus was on full-scale framework application, with multiple participants in a GDSP. The project had 18 participants, 4 placed in Finland and 14 in Denmark. The workshop included all the Finnish participants and ten of the Danish participants and took place by the end of the requirements specification phase. The workshop was divided into three activities over a two-day period. The first activity, individual risk assessments, was conducted the first day with the DPMS tool and facilitated by one of the authors through individual 15-min sessions with each participant. The second activity consisted of following the remaining steps in the risk management framework. First, each risk area was discussed and reassessed based on the presentation of the individual assessments. Second, the risk area with the highest risk exposure was chosen, and appropriate resolution techniques for that specific risk area were selected and planned. The third and final activity was a group debate of how the participants experienced the framework.

The overall assessment was that the framework was useful in facilitating awareness among project participants of important challenges related to geographical distribution. The four hours each participant used on the framework was considered well spent. During the discussions, the participants spent a significant amount of time on making sense of and applying the concepts to their specific project context. This was necessary in order to clarify project roles across participants and sites and to exchange the different frames of reference caused by geographical distribution of the project. These discussions were considered at least as valuable as the resulting planning documents. The individual assessments were pointed out as an important prerequisite for a valuable discussion. The 15 min allocated for individual assessment was, however, deemed to be too little time; it was estimated that 45 min on average was required for individual assessments. The participants pointed out, that the Web-based tool used risk aversive language in describing overall levels of risk factors. Instead of indicating an "unsatisfactory" risk level we relabeled it to "high." The changes in the framework resulting from Evaluation IV led to the current version of the framework.

VII. Discussion and Implications

GDSPs are becoming increasingly common. This trend is driven by global competition, increased need for flexibility, access to global resources, and substantial financial gains [37], [91]; moreover, available electronic communication infrastructures make GDSPs increasingly feasible to organize and manage [97], [101]. Although GDSPs have developed into a fertile area of research, there are no comprehensive management frameworks available for this type of organization [25], [91]. Our study took steps toward filling this void by applying risk management [4], [7], [39], [67], [75] to review and synthesize the state-of-the-art knowledge on GDSPs (e.g., [6], [37], [91], [99], [108]). As a result, we developed new risk concepts and a risk management framework, as summarized in Fig. 5.

Specifically, we carried out the research objective to "integrate the existing knowledge into a practically useful framework for managing risks inherent in GDSPs" through two activities. First, we synthesized current knowledge into conceptualizations of risks and resolution techniques and integrated these into a risk-action list framework [7], [39] for GDSPs. The framework consists of three primary elements: risk assessment, risk resolution, and risk management planning, and it

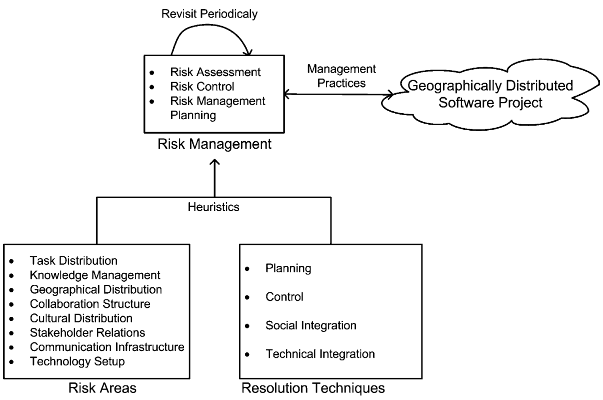

Fig. 5. Summary of research.

provides heuristics for applying four types of resolution techniques to eight risk areas (Table IV). The synthesizing activity drew on the existing literature to help develop clear and distinct conceptions of both risks and resolution techniques. We used available surveys of challenges in GDSPs to identify relevant risk areas consistent with the state-of-the-art knowledge (Table II); used Leavitt's systems model [60], as proposed by Lyytinen *et al.* [67], to clarify the primary focus of the identified risk areas (Table II); aggregated a complete list of risk factors identified in the literature and categorized them according to the proposed risk areas; and finally, further characterized each risk area through detailed definitions of the involved risk factors (Table III). Concerning risk resolution techniques, we adapted McFarlan's [75] well-established categories of project management techniques to distributed contexts (Table I). Finally, concerning heuristics for applying resolution techniques to risk areas, we based our framework on systematically applying the state-of-the-art research on GDSPs (Table IV).

Second, we revised the framework iteratively based on evaluations of its practical usefulness [81]. To that end, we drew on a variety of empirical evaluation methods. Initially, we had experienced how software practitioners assess the relevance and understandability of both risk and risk resolution concepts through two focus groups (Table V). Three subsequent evaluations aimed at assessing the practical use of the framework through workshops with experienced practitioners (Table V); to support the last two workshops, we implemented a Web-based tool, DPMS, thereby making the framework readily available for practical use. In conclusion, the framework was found to be easy to use and to provide relevant support for managing the projects under consideration. Section VI describes the evaluation activities and the iterative changes to the framework (Table V). Throughout the paper, we have consistently presented the final version of the framework.

The framework is, in this way, based on systematic synthesis of the literature and systematic evaluation of practical usefulness. Still, the presented research has notable limitations. First, we have provided only preliminary evaluations of the practical usefulness of the proposed framework. There is a need for more research into the utility of the framework across different types of contexts. Second, the set of 72 articles from which the synthesis of risks, resolution techniques, and heuristics are derived is limited due to the adopted criteria for selecting literature; we acknowledge the existence of additional articles, books, and other sources, which could potentially contribute to risk management for GDSPs. Third, the participants in our evaluations were of Danish, Spanish, and Finnish origin; this means that cultural challenges of framework usage, e.g., potential differences in interpretation of explanatory texts, rigor of framework application, or different perceptions of project management, need further investigation.

The integrative framework makes a contribution to the literature on GDSPs. It is, to our knowledge, the first comprehensive framework for managing risks in this increasingly important type of software project. The framework provides an overview and synthesis of the state-of-the-art knowledge, provides conceptualizations of risks and resolution techniques related to GDSPs, and opens for a number of interesting research opportunities. A combination of action research and case studies could investigate practical, long-term effects on GDSPs and interactions with the organizational and cultural context and with other managerial control mechanisms [53], [54]; address the recent call for research on control in distributed environments [54], [91]; and investigate the effects on knowledge sharing and cohesion in GDSPs. Future experiments could compare and contrast differences in management behavior between subjects with and without risk management support and test for differences across contexts supported by the overall project risk measures published by Barki *et al.* [4] or Keil [49]. Finally, our research calls for further investigations into available resolution techniques. Such studies could help us learn more about resolution techniques by investigating how effectively they support risk management decisions and managerial strategies in different contexts. Such studies could also draw on the schema of heuristics between risk areas and resolution techniques (Table IV) and adopt design science research to develop and combine techniques to address more risk areas.

The framework also has important implications for management practices within the software industry. The framework demonstrates the complex nature of risks in GDSPs, and it offers concepts and heuristics that practitioners can use to assess and control the risks they face in specific projects. The framework can be used by project managers and participants at any stage of a GDSP, but it is recommended to revisit risk management regularly during a project's lifetime, as illustrated in Fig. 5. On one level, the framework provides participants in GDSPs with a vocabulary they can use to identify, reflect on, and share management problems and solutions [8], [102]. This vocabulary is summarized in Fig. 5. On another level, the framework provides support for assessing risks (Table III) and identifying appropriate resolution techniques (Table IV). Practitioners are generally advised to go through the steps of risk assessment, risk control, and risk management planning, as summarized in Fig. 2 and described in detail in Section V; one possible implementation of the framework as a Web-based tool is made available at http://www.distributedprojects.net.

Appendix A: Journal List

Academy of Management Journal
Academy of Management Review
ACM Computing Surveys
ACM Special Interest Group Publications
ACM Transactions (various)
Administration and Society
Administrative Science Quarterly
AI Magazine
American Journal of Sociology
American Psychologist
American Sociological Review
Arbitration Journal
Artificial Intelligence
Australian Journal of Information Systems
California Management Review
Communications of the ACM
Communications of the AIS
Computer Journal
Computer Supported Cooperative Work
Computers and Operations Research
Database
Database Programming and Design
Decision Sciences
Decision Support Systems
European Journal of Information Systems
Expert Systems with Applications
Harvard Business Review
Human Relations
Human Resource Management
Human-Computer Interaction
IBM Systems Journal
IEEE Computer
IEEE Software
IEEE Transactions (various)
Industrial and Labor Relations Review
Industrial Relations
Information & Management
Information and Organization (formerly Accounting, Management, and IT)
Information and Software Technology
Information Resources Management Journal
Information Science
Information Systems
Information Systems Journal
Information Systems Management
Information Systems Research
Interfaces (INFORMS)
International Journal of Human-Computer Studies
International Journal of Information Management
International Journal of Man-Machine Studies
International Journal of Technology Management
Journal of Applied Behavioral Science
Journal of Applied Psychology
Journal of Business Research
Journal of Business Strategy
Journal of Collective Negotiations in the Public Sector
Journal of Computer and System Sciences
Journal of Computer Information Systems
Journal of Conflict Resolution
Journal of Database Administration
Journal of Education for Management Information Systems
Journal of Engineering and Technology Management
Journal of General Management
Journal of Global Information Management
Journal of Global Information Technology Management
Journal of Human Resources
Journal of Information Management
Journal of Information Science
Journal of Information Systems (accounting)
Journal of Information Systems (education)
Journal of Information Systems Management
Journal of Information Technology
Journal of Information Technology Management
Journal of International Business Studies
Journal of International Information Management
Journal of Management
Journal of Management Information Systems
Journal of Management Studies
Journal of Management Systems
Journal of Occupational Psychology
Journal of Organizational Behavior
Journal of Personality and Social Psychology
Journal of Strategic Information Systems
Journal of Systems and Software
Journal of Systems Management
Journal of the ACM
Journal of the AIS
Journal of the American Society for Information Science
Journal of Vocational Behavior
Journal on Computing
Knowledge Based Systems
Labor Law Journal
Long Range Planning
Management Science
MIS Quarterly
MISQ Discovery
Monthly Labor Review
Omega
Operations Research
Organization Science
Organizational Behavior and Human Decision Processes
Organizational Dynamics
Organizational Studies
Personnel Psychology
Psychological Bulletin
Psychological Review
Research in Organizational Behavior
Scandinavian Journal of Information Systems
Sloan Management Review
Social Forces
Strategic Management Journal
WIRT (Wirtschaftsinformatik)

APPENDIX B:
REVIEWED ARTICLES

Journal	Article	Ref.
IEEE TRANSACTIONS ON PROFESSIONAL COMMUNICATION (7)	Panteli, N; Davison, RM (2005)	[84]
	Sarker, S; Sarker, S; Nicholson, DB; Joshi, KD (2005)	[101]
	Ocker, RJ (2005)	[82]
	Bradner, E; Mark, G; Hertel, TD (2005)	[9]
	Rutkowski, AF; Vogel, DR; Van Genuchten, M; Bemelmans, TMA; Favier, M (2002)	[97]
	Vogel, DR; van Genuchten, M; Lou, D; Verveen, S; van Eekout, M; Adams, A (2001)	[110]
	Tan, BCY; Wei, KK; Huang, WW; Ng, GN (2000)	[105]
IEEE SOFTWARE (6)	Carmel, E; Agarwal, R (2001)	[12]
	Herbsleb, JD; Moitra, D (2001)	[37]
	Ebert, C; De Neve, P (2001)	[24]
	Heeks, R; Krishna, S; Nicholson, B; Sahay, S (2001)	[34]
	Battin, RD; Crocker, R; Kreidler, J; Subramanian, K (2001)	[5]
	Herbsleb, JD; Grinter, RE (1999)	[35]
JOURNAL OF MANAGEMENT INFORMATION SYSTEMS (5)	Paul, S; Samarah, IM; Seetharaman, P; Mykytyn, PP (2004)	[85]
	Pauleen, DJ (2003)	[87]
	Massey, AP; Montoya-Weiss, MM; Hung, YT (2003)	[70]
	Kayworth, TR; Leidner, DE (2001)	[47]
	Dennis, AR; Wixom, BH (2001)	[19]
COMMUNICATIONS OF THE ACM (4)	Dubé, L; Paré, G (2001)	[22]
	Sabherwal, R (1999)	[98]
	Taylor, H (2006)	[106]
	Krishna, S; Sahay, S; Walsham, G (2004)	[56]
INFORMATION & MANAGEMENT (4)	Paul, S; Seetharaman, P; Samarah, I; Mykytyn, PP (2004)	[86]
	Lander, MC; Purvis, RL; McCray, GE; Leigh, W (2004)	[58]
	Larsen, KRT; McInerney, CR (2002)	[59]
	Lurey, JS; Raisinghani, MS (2001)	[64]
ORGANIZATION SCIENCE (4)	Hinds, PJ; Bailey, DE (2003)	[38]
	Cramton, CD (2001)	[17]
	Maznevski, ML; Chudoba, KM (2000)	[73]
	Jarvenpaa, SL; Leidner, DE (1999)	[41]
INFORMATION SYSTEMS JOURNAL (3)	Chudoba, KM; Wynn, E; Lu, M; Watson-Manheim, MB (2005)	[16]
	Furst, S; Blackburn, R; Rosen, B (1999)	[30]
	Warkentin, M; Beranek, PM (1999)	[111]
ACADEMY OF MANAGEMENT JOURNAL (2)	Montoya-Weiss, MM; Massey, AP; Song, M (2001)	[77]
	Kirkman, BL; Rosen, B; Tesluk, PE; Gibson, CB (2004)	[52]
IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT (2)	Lee, JN; Kim, YG (2005)	[61]
	McDonough, EF; Kahn, KB; Griffin, A (1999)	[74]
INFORMATION AND SOFTWARE TECHNOLOGY (2)	Jacobs, J; van Moll, J; Krause, P; Kusters, R; Trienekens, J; Brombacher, A (2005)	[40]
	Sakthivel, S (2005)	[99]
INFORMATION SYSTEMS RESEARCH (3)	Majchrzak, A; Malhotra, A; John, R (2005)	[68]
	Jarvenpaa, SL; Shaw, TR; Staples, DS (2004)	[42]
	Choudhury, V; Sabherwal, R. (2003)	[15]
JOURNAL OF INFORMATION TECHNOLOGY (2)	Breu, K; Hemingway, CJ (2004)	[10]
	Klepper, R (1995)	[55]
	Pauleen, DJ; Yoong, P (2001)	[88]
ORGANIZATIONAL DYNAMICS (2)	Zigurs, I (2003)	[117]
	Cascio, WF; Shurygailo, S (2003)	[13]
ACADEMY OF MANAGEMENT EXECUTIVE (1)	Townsend, AM; DeMarie, SM; Hendrickson, AR (1998)	[108]
DATABASE FOR ADVANCES IN INFORMATION SYSTEMS (1)	Powell, A; Piccoli, G; Ives, B (2004)	[91]
DECISION SCIENCES (1)	Warkentin, ME; Sayeed, L; Hightower, R (1997)	[112]
EUROPEAN JOURNAL OF INFORMATION SYSTEMS (1)	Sarker, S; Sahay, S (2004)	[100]
EUROPEAN MANAGEMENT JOURNAL (1)	Kayworth, T; Leidner, D (2000)	[46]
GROUP & ORGANIZATION MANAGEMENT (1)	Bell, BS; Kozlowski, SWJ (2002)	[6]
HARVARD BUSINESS REVIEW (1)	Majchrzak, A; Malhotra, A; Stamps, J; Lipnack, J (2004)	[69]
HUMAN RESOURCE MANAGEMENT (1)	DeRosa, DM; Hantula, DA; Kock, N; D'Arcy, J (2004)	[20]
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING (1)	Herbsleb JD, Mockus A (2003)	[36]
INDUSTRIAL MANAGEMENT & DATA SYSTEMS (1)	van der Smagt, T (2000)	[109]
INFORMATION AND ORGANIZATION (1)	Nicholson, B; Sahay, S (2001)	[80]
INFORMATION SYSTEMS (1)	Farshchian, BA (2001)	[28]
INFORMATION SYSTEMS MANAGEMENT (1)	Ramarapu, N; Parzinger, MJ; Lado, AA (1997)	[93]
INTERNATIONAL JOURNAL OF CONFLICT MANAGEMENT (1)	Mortensen, M; Hinds, PJ (2001)	[78]
INTERNATIONAL JOURNAL OF HUMAN-COMPUTER STUDIES (1)	Potter, RE; Balthazard, PA (2002)	[90]
INTERNATIONAL JOURNAL OF INFORMATION MANAGEMENT (1)	Khalfan, AM (2004)	[51]
JOURNAL OF AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY (1)	Alavi, M; Tiwana, A (2002)	[1]
JOURNAL OF COMPUTER INFORMATION SYSTEMS (1)	Xue, YJ; Sankar, CS; Mbarika, VWA (2004)	[116]
JOURNAL OF ENGINEERING AND TECHNOLOGY MANAGEMENT (1)	Evaristo, JR; Scudder, R; Desouza, KC; Sato O (2004)	[25]
JOURNAL OF MANAGEMENT (1)	Shin, Y (2004)	[103]
JOURNAL OF STRATEGIC INFORMATION SYSTEMS (1)	Kanawattanachai, P; Yoo, Y (2002)	[44]
MIS QUARTERLY (1)	Piccoli, G; Ives, B (2003)	[89]
ORGANIZATIONAL BEHAVIOR AND HUMAN DECISION PROCESSES (1)	Alge, BJ; Wiethoff, C; Klein, HJ (2003)	[2]
PERSONNEL JOURNAL (1)	Solomon, CM (1995)	[104]
PROCEEDINGS OF THE 35TH ANNUAL HAWAII INTERNATIONAL CONFRENCE ON SYSTEM SCIENCES (1)	Ramesh, V ; Dennis, AR (2002)	[94]

APPENDIX C: RESOLUTION TECHNIQUES

Planning

Acquire complementary skills

- Involve representatives from other sites in recruitment of new project participants [15].

Important qualities and competencies consist of:

- Patience, stamina, and persistence [112].
- Appreciation of autonomy, flexibility, and diversity [103], [112].
- Trustfulness, reliability, lateral abilities, and skills in virtual communication [103], [108].
- Domain knowledge, computer literacy, and skills in time coordination [10], [103].
- Highly developed collaboration skills [108].
- Understanding of different cultures [108].
- Experience in mediated collaboration [20].
- Homogeneity with respect to culture and education [56], [78].

Adjust meetings to distributed context

- Identify subjects of the meeting subsequently to determine the relevant participants [100], [112].
- Determine the requirements of the meeting. If visual aids are necessary, a videoconference may be appropriate [13].
- Ensure well-prepared participants by sending agenda and other important documents prior to the meeting. Participants should additionally be informed if they are expected to have material ready for the meeting [13], [112].
- Elect a presiding officer (PO) of the meeting to ensure appropriate turn taking and compliance of the agenda [112].
- The PO should summarize discussions periodically during meetings [22], [56].
- The PO should keep an attendance register and ask the participants to introduce themselves at the beginning of the meeting as well as upon speaking [13].
- In the case of a telephone conference: The PO should describe activities in the room, people coming and going, and drawn objects [13].
- Alleviate drawbacks, i.e. in terms of odd work hours, of time zone differences by using fixed meeting schedules. Additionally by "sharing the pain", distributing drawbacks equally among participants [13], [88], [100].

Divide tasks systematically between sites

- Focus on initial analysis of the modular structure to identify dependencies and predict the effects of changes [36].
- Distribute tasks prioritized according to placement of expertise and local advantages, e.g., proximity to customer [5].
- Base the task distribution on a well-devised modular design that takes the project structure into account. Segmented tasks should be thoroughly understood and expectedly stable [35], [36].
- Keep development and maintenance separate [24].
- Segment the task based on different user requirements [24].
- Segment the system horizontally in clusters with end-to-end functionality. Each group has the responsibility for any and all aspects of functions across system layers in the appointed cluster [5].
- Define an architecture "light" with few, but important principles. The architecture should describe the entire system on a high level of abstraction [5].
- Segment the system sequentially, such that each element is parsed on only upon completion, e.g., from development to test [24], [36].

Reduce coupling between sites

- Reducing dependencies between sites can be done in two ways: Provide well-defined work areas and thereby give the participants minimal control, or give extensive responsibility and thereby full participant control [12].
- Use object-oriented groups. As with object-oriented programming, tasks are split with well-defined interfaces allowing for separation of the groups [91], [94].
- Combine with: *Divide tasks systematically between sites* [5].

Create shared collaboration platform

- Introduce a shared vocabulary to describe both everyday activities in the organization and central activities in the development process, thereby promoting unity and sense of belonging and reducing misinterpretations, e.g., using UML [5], [16], [100].
- Use dialogue technique to establish shared mental models of the organization and tasks. The dialogue technique is further elaborated by Tan *et al.* [105].
- Produce lists of concepts, explaining slang expressions of the involved cultures and professions [69].
- Utilize a shared tool to track bugs and corrections in every part of the distributed organization, thereby providing transparency [5].
- Establish shared project culture without discriminating in favor of any particular national or professional culture [WS1].
- Combine with: Establish shared goals [24], [46].

Establish shared goals

- The project manager should explicitly relate the distributed project organization to the overall strategy, mission and vision, and communicate the purpose of the task [10], [15], [52], [58].
- The project manager should establish and obligate the participants to a shared project goal to develop a common identity [6], [24], [30], [46], [98], [104].
- The project manager should act both proactively and creatively by periodically updating participants about progress towards the planned goals [46].
- Institute a program of implementation that puts focus on the project contribution and relates it to the participants' present qualities and contribution to the company and its products [108].
- Combine with: *Improve Distributed collaboration skills* and *Improve collaboration and communication technology skills* [111].

Establish communication norms

- The project manager should institute time-based norms for communication, virtual presence, and development of solidarity to encourage tolerance of silences and delayed responses [100].

- Create formal guidelines for the use of asynchronous media that include descriptions of what the communication should contain and how it should be performed [88], [111].
- Set up rules for carrying out written communication, e.g., during chat sessions. Rules should include the use of sarcasm, jokes, the use of capital letters. Ensure that language use is not misinterpreted [111], [117].
- Include contextual information in the communications, such as name, title, and position to facilitate the development of social relations [13], [117].
- Establish conventions for answering messages in, e.g., instant messaging or email. This could include using the 'reply' function or explicitly naming receiver [47], [100], [112].
- Create clear policies for personal privacy [108].

Define roles and responsibilities

- At project startup, define undisputed areas of responsibility for all participants as well as the relational roles being instituted [24], [41], [46], [47], [51], [117].
- The project manager should act as role model and demonstrate the efficiency, quality, and skills needed to create mutual respect in the distributed environment [59].
- The project manager should exercise authority to ensure the completion of delegated assignments [46], [47].

Reduce time-zone differences

- Choose remote sites in the same or proximate time zones, thus alleviating time distance [12], [99].

Control

Focus on deliverables

- Plan and control critical deliverances perfectly considering differences in, e.g. tools and data format [22], [37].

Establish task coordination between sites

- Adjust the project structure dynamically to the individual phases of the project to avoid bottlenecks and waste of resources [97].
- Define clear-cut criteria for both beginning and end of the different project phases [37].
- Use incremental integration of task segments, and avoid the 'big bang' at the end of the project [5].

Maintain site autonomy

- Make sites self-managing by establishing a system that allows participants to monitor own processes [6], [25], [55].
- Be careful using traditional mechanisms for behavioral control as well as managerial interventions as focus on deadlines and progress can lead to mistrust [89].
- Introduce team empowerment [5], [52].

Establish shared control mechanisms

- Design a method to monitor and control the information flow concerning project status, providing information to all stakeholders including the individual groups [25], and accept the administrative overhead [WS1].
- Let participants themselves monitor changes in their environment [6].
- The project manager should make sure that periods of inactivity are detected and reported automatically [110].
- Deploy tools to measure trust in the project's early stages [44].
- Emphasize focus on results and performance compared to traditional projects [20], [104].

Establish temporal coordination mechanisms

- Use structured mechanisms for temporal coordination, including handling of deadlines, synchronization, and distribution of resources [70], [77].
- Introduce shared deadlines or milestones when coordinating successive integration of individual software modules as well as handling diversities concerning local festivals and holidays [5], [59], [100].
- If reduction of temporal distance is impossible: The project manager should manage time translations and time adjustments, relocate time using asynchronous media, and institute time-based norms for communication and virtual presence [12], [99].
- Focus on synchronization, plans, and procedures in the development process, enabling transferring of tasks from one site to another [15], [99].

Maintain project organization overview

- Make the lines of communication as short as possible [74].
- Create a clear plan of communication based on individual and group communication needs. The plan should further contain a list of contacts and their professional and decision making competencies [35].
- Create a database that contains the areas of expertise of the individual project participants [28], [36], [100].
- Use IT systems such as instant messaging to determine availability of participants [36].
- Make internal group decision processes transparent by exchanging meeting résumés between sites [100].
- Combine with: *Define roles and responsibilities* [24], and *Adopt appropriate communication technologies* [36].

Maintain task overview within and across sites

- Define an architecture "light" with few, but important principles. The architecture should describe the entire system on a high level of abstraction [5].
- At project startup: Define clearly, which groups are involved in the project, and which tasks they work on [68].
- Create a project website that summarizes project content, progress, planning, and group related information [68].
- Use dialogue technique to establish shared mental models of the organization and tasks. The dialogue technique is further elaborated in Tan *et al.* [105].
- Utilize a model (software system), which supports consciousness about the development at remote sites, focusing on which information should be conveyed to whom [28].
- Use a content management system (CMS) [WS1].
- Evaluate the task complexity considering task structure, task environment, and internal and external coupling [6].

Monitor and improve communication

- Intervene often and rapidly in the communication and provide frequent feedback to the participants [47].
- The project manager should intervene rapidly if there is a minimal risk of a problem not being resolved [110].
- Provide tools and strategies for early conflict management [44].

- The project manager should control the amount and the quality of the mediated communication [88].

Maintain a supportive environment

- Use a supportive non-dictating management style [110].
- Express flexibility and empathy towards the participants [46], [58].
- The project manager should act as roll model and demonstrate the efficiency, quality, and skills needed to create mutual respect in the distributed environment [59], [100].
- The management style should not be characterized by surveillance [59].
- Use “Collaborative conflict management style” (high interest in other’s opinion and high interest in own opinion) [85], [86].
- Avoid creativity inhibiting factors, e.g., time pressure and overly firm structure [82].
- Create heterogeneous groups [82].
- The project manager should set a good example, producing creative ideas — especially in the startup phase [82].
- Prioritize iterative processes in both problem specification and design [82].
- Choose ICT that supports a decentralized network between participants, enhancing information flow and generation of new ideas [97].
- Combine with: *Create a shared collaboration platform* [82] and *Increase technical compatibility between sites* [82].

Analyze and manage errors

- Jacobs *et al.* [40] present a long list of potential causes for errors and delays. We find this too extensive to be presented here. No particular resolution technique is presented in the article, but the individual entries in the list are in our opinion sufficiently specific to be converted to such.

Social Integration

Improve capability to manage cultural differences

- Establish courses in cultural diversity during the startup phase of the project. If participants are stationed at remote sites, the cultural training should take place before departure [22], [56].
- Focus on creating understanding and acceptance of differences, e.g., by letting each participant make a presentation on their individual culture, values, and expectations [22], [104].
- Promote understanding and acceptance rather than seek to streamline the project organization [WS1].
- Focus on the strengths that diversity offers rather than the weaknesses [22], [100].
- Acknowledge and discuss cultural differences in a respectful and civilized manner [104], and keep in mind that there are limits to cultural adaptation [56].
- Adjust management style according to culture, e.g., participants’ preferences for well-defined tasks vs. preference for loosely defined task and self-management [WS1].
- Combine with: *Promote humor and openness* [47], [91].

Improve distributed collaboration skills

- Educate and train participants in collaboration skills specific for the distributed environment [13], [22], [30], [85], [91], [100], [104], [108], [111], [117]. Skills are divided into three main areas: Virtual collaboration skills, virtual communication skills, and virtual socialization skills [13].
- Focus on creating task- and group related processes [30], [108].
- Offer training rather than relying on localized best practices [117].
- Conflict management should be part of the training [85].
- Seek to obtain a dialogue rather than a two-way monologue in the communication [109].
- Combine with: *Improve capability to manage cultural differences* [104].

Improve language skills

- Introduce language training [24], [88].
- Establish English as the official language of the organization and introduce language training (if collaboration is intra-organizational) [24].
- Use supporting technologies, such as spell checkers and translators [22].

Emphasize early teambuilding activities

- Stimulate the interaction between participants already from project startup [46], [58], [98], [110].
- Arrange videoconferences if face-to-face meetings are impossible [97], [110].
- Create cross functional groups in the initial phase of the project to encourage social relations across areas of expertise [24].
- Combine with: *Use mentors to integrate new members* [105].

Promote humor and openness

- Stimulate relationship building by using humor [47], [91].
- Use humor to enhance cultural understanding by making fun of differences instead of ignoring them [104].
- Promote openness to avoid hidden agendas that impede productivity [WS1].

Use mentors to integrate new members

- Use mentors to integrate new participants. The mentor is responsible for social adaptation and communication of the project and group’s history and values [6], [30], [103], [105].

Use face-to-face meetings appropriately

- Prioritize face-to-face meetings to develop trust and shared identity easier and faster [30], [34], [35], [64], [87], [91], [117].
- Use the “Sandwich structure”, i.e., hold face-to-face meetings at the beginning and at the end of the project. The first to create relationships and trust, and the second to summarize experiences as well as ensure that everybody can look each other in the eye [35], [97], [110], [112].
- The project manager should travel to all distributed sites and keep closest to the sites where task coupling is high and group cohesion low [99].
- Plan meetings using a simple- or double rhythm. Using simple rhythm, meetings are held at a specified interval. Double rhythm consists of one meeting at project startup and another midway to summarize problems [73].

Develop liaisons between sites

- Use liaisons to facilitate information exchange, identify expertise, mediate cultural conflicts, and settle disputes [5], [12], [24], [34–36], [56] [WS1].
- If the organization consists of a main site and several sub sites: Let liaisons from the sub sites spend the startup phase at the main site to gain insight and overview of the project [5].
- Include travel expenses in the overall budget and do not perceive them as additional costs [WS1] [34].

Adopt shared reward systems

- Deploy evaluations and reward structures, which encourage group related behavior, to create cohesion - individual rewards are not advised [10], [30], [64], [108], [109] [WS1].

Technical Integration

Increase technical compatibility between sites

- Ensure that sites are technically compatible [22], [37].
- Choose sites where it is possible to maintain high quality transmissions at low cost considering parameters such as the country's technological infrastructure and different laws and regulations [10], [22], [46].
- Make international support contracts with sub contractors to ensure access to support at all sites [5].

Standardize and train in methods across sites

- Establish technical standards and stick to them to maintain project integrity [25].
- Introduce shared guidelines for error handling, accessibility to other group's documentation, and documentation for tests and testability [40].
- Use a shared tool that allows for tracking of corrections and bugs in all parts of the distributed organization [5].
- Standardize tools, methods and processes to create a harmonic and efficient project organization [24], [25], [56], [99].
- Such standardization implies training of participants and lower initial efficiency as experience with the chosen methods varies [5], [25], [46]. In the long run higher efficiency and fewer misunderstandings are expected [24], [56].
- Combine with: *Combine waterfall model and prototyping* [99].

Adopt appropriate communication technologies

- Include the country's infrastructure and bandwidth possibilities when considering choice of communication technology [10], [22], [46].
- Choose rich media if participants belong to a high context culture, where message interpretation relies heavily on contextual information. If participants belong to a low-context culture, leaner media can be used [88].
- Include speed and quality in the communication media requirements as well as stability and reliability [17], [22], [46], [116].
- Choose ICT that supports a suitable atmosphere for creation of trust, e.g., videoconferences can reduce the sense of physical and psychological distance [44].
- Support creativity by choosing ICT that supports a decentralized communication network, e.g., instant messaging [36], [97], [101].
- Utilize videoconference or telephone conference in addition to group support systems usage, whenever the group is working on a decision making task [19].
- Use telephone- and videoconferences to support relationship building [64].
- Consider cultural preferences when choosing ICT [88].
- Choose as rich media as possible to support social processes, collaboration, and cohesion [1], [30], [47], [86], [110], [111].
- Choose lean media, such as e-mail, for simple messages to reduce noise in the information. For complex and debatable information, choose richer media [68], [110].
- If the project has a large distribution over time zones: Consider asynchronous media [97].
- Larger projects (in terms of number of participants) should adopt communication technology that supports coordination and logistics, whereas smaller projects should choose technology that supports collaboration and communication [9].

Improve collaboration and communication technology skills

- Focus on strengthening the participants' collaboration and communication skills [5], [30], [46], [91], e.g., training in a wide variety of technologies [10], [46], [91].
- Use ICT training to enhance the use of lean media and thereby making them richer [111].
- Train participants in sharing contextual and social information [68], [91].

Improve development technology skills

- Educate and train in software development technology [22], [46], [100], especially participants being introduced to new technology [46], [56].
- Standardize training across sites [91], [100].

Handle differences in methods between sites

- The alternative to standardization: Manage differences by taking advantage of existing expertise, create fast results, and avoid expenses from training and adaptation to new methods [5], [25], [100].
- Communicate knowledge of methods and processes deployed by other groups and enhance understanding of strengths and weaknesses [100].
- "Capability Maturity Model" (CMM) can be used to ensure equal quality level of the deferring methods [25].
- Combine with: Divide Tasks Systematically Between Sites and Reduce Coupling Between Sites [5].

Combine waterfall model and prototyping

- Combine the waterfall model and prototyping, using prototyping to determine requirements and the waterfall model to maintain modularity, low task coupling, and structure in the process [99].

REFERENCES

[1] M. Alavi and A. Tiwana, "Knowledge integration in virtual teams: The potential role of KMS," *J. Amer. Soc. Inf. Sci. Tec.*, vol. 53, no. 12, pp. 1029–1037, 2002.

[2] B. J. Alge, C. Wiethoff, and H. J. Klein, "When does the medium matter? Knowledge-building experiences and opportunities in decision-making teams," *Organ. Behav. Hum. Decis. Process.*, vol. 91, no. 1, pp. 26–37, 2003.

[3] S. Alter and M. Ginzberg, "Managing uncertainty in MIS implementation," *Sloan Manage. Rev.*, vol. 20, no. 1, pp. 23–31, 1978.

[4] H. Barki, S. Rivard, and J. Talbot, "Toward an assessment of software development risk," *J. Manage. Inf. Syst.*, vol. 10, no. 2, pp. 203–225, 1993.

[5] R. D. Battin, R. Crocker, J. Kreidler, and K. Subramanian, "Leveraging resources in global software development," *IEEE Softw.*, vol. 18, no. 2, pp. 70–77, Mar./Apr. 2001.

[6] B. S. Bell and S. W. J. Kozlowski, "A typology of virtual teams—Implications for effective leadership," *Group Organ. Manage.*, vol. 27, no. 1, pp. 14–49, 2002.

[7] B. W. Boehm, "Software risk management—Principles and practices," *IEEE Softw.*, vol. 8, no. 1, pp. 32–41, Jan. 1991.

[8] R. J. Boland, Ed. (2002), "Design in the punctuation of management action," in *Managing as Designing: Creating a Vocabulary for Management Education and Research*, Frontiers of Management Workshop, Weatherhead School of Management [Online]. Available: http://www.design.cwru.edu.

[9] E. Bradner, G. Mark, and T. D. Hertel, "Team size and technology fit: Participation, awareness, and rapport in distributed teams," *IEEE Trans. Prof. Commun.*, vol. 48, no. 1, pp. 68–77, Mar. 2005.

[10] K. Breu and C. J. Hemingway, "Making organisations virtual: The hidden cost of distributed teams," *J. Inf. Technol.*, vol. 19, no. 3, pp. 191–202, 2004.

[11] R. N. Burns and A. R. Dennis, "Selecting the appropriate application development methodology," *Data Base*, vol. 17, no. 1, pp. 19–23, 1985.

[12] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," *IEEE Softw.*, vol. 18, no. 2, pp. 22–29, Mar./Apr. 2001.

[13] W. F. Cascio and S. Shurygailo, "E-leadership and virtual teams," *Organ. Dyn.*, vol. 31, no. 4, pp. 362–376, 2003.

[14] R. N. Charette, *Software Engineering Risk Analysis and Management*. New York: McGraw-Hill, 1989.

[15] V. Choudhury and R. Sabherwal, "Portfolios of control in outsourced software development projects," *Inf. Syst. Res.*, vol. 14, no. 3, pp. 291–314, 2003.

[16] K. M. Chudoba, E. Wynn, M. Lu, and M. B. Watson-Manheim, "How virtual are we? Measuring virtuality and understanding its impact in a global organization," *Inf. Syst. J.*, vol. 15, no. 4, pp. 279–306, 2005.

[17] C. D. Cramton, "The mutual knowledge problem and its consequences for dispersed collaboration," *Organ. Sci.*, vol. 12, no. 3, pp. 346–371, 2001.

[18] G. B. Davis, "Strategies for information requirements determination," *IBM Syst. J.*, vol. 21, no. 1, pp. 4–30, 1982.

[19] A. R. Dennis and B. H. Wixom, "Investigating the moderators of the group support systems use with meta-analysis," *J. Manage. Inf. Syst.*, vol. 18, no. 3, pp. 235–257, 2001.

[20] D. M. DeRosa, D. A. Hantula, N. Kock, and J. D'Arcy, "Trust and leadership in virtual teamwork: A media naturalness perspective," *Hum. Resour. Manage.*, vol. 43, no. 2–3, pp. 219–232, 2004.

[21] S. E. Donaldson and S. G. Siegel, *Successful Software Development*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[22] L. Dube and G. Pare, "Global virtual teams," *Commun. ACM*, vol. 44, no. 12, pp. 71–73, 2001.

[23] M. Earl, *Information Management Strategy*. Englewood-Cliffs, NJ: Prentice-Hall, 1987.

[24] C. Ebert and P. D. Neve, "Surviving global software development," *IEEE Softw.*, vol. 18, no. 2, pp. 62–69, Mar./Apr. 2001.

[25] J. R. Evaristo, R. Scudder, K. C. Desouza, and O. Sato, "A dimensional analysis of geographically distributed project teams: A case study," *J. Eng. Technol. Manage.*, vol. 21, no. 3, pp. 175–189, 2004.

[26] R. Evaristo and P. C. van Fenema, "A typology of project management: Emergence and evolution of new forms," *Int. J. Proj. Manage.*, vol. 17, no. 5, pp. 275–281, 1999.

[27] R. Fairley, "Risk management for software projects," *IEEE Softw.*, vol. 11, no. 3, pp. 57–67, May/Jun. 1994.

[28] B. A. Farshchian, "Integrating geographically distributed development teams through increased product awareness," *Inf. Syst.*, vol. 26, no. 3, pp. 123–141, 2001.

[29] R. H. Franke, T. W. Edlund, and F. Oster, "The development of strategic management–Journal quality and article impact," *Strateg. Manage. J.*, vol. 11, no. 3, pp. 243–253, 1990.

[30] S. Furst, R. Blackburn, and B. Rosen, "Virtual team effectiveness: A proposed research agenda," *Inf. Syst. J.*, vol. 9, no. 4, pp. 249–269, 1999.

[31] J. R. Galbraith, *Designing Complex Organizations*, 1st ed. Boston, MA: Addison-Wesley Longman, 1973.

[32] M. J. Ginzberg, "Early diagnosis of MIS implementation failure—Promising results and unanswered questions," *Manage. Sci.*, vol. 27, no. 4, pp. 459–478, 1981.

[33] L. R. Gomezmejia and D. B. Balkin, "Determinants of faculty pay—An agency theory perspective," *Acad. Manage. J.*, vol. 35, no. 5, pp. 921–955, 1992.

[34] R. Heeks, S. Krishna, B. Nicholson, and S. Sahay, "Synching or sinking: Global software outsourcing relationships," *IEEE Softw.*, vol. 18, no. 2, pp. 54–60, Mar./Apr. 2001.

[35] J. D. Herbsleb and R. E. Grinter, "Architectures, coordination, and distance: Conway's law and beyond," *IEEE Softw.*, vol. 16, no. 5, pp. 63–70, Sep./Oct. 1999.

[36] J. D. Herbsleb and A. Mockus, "An empirical study of speed and communication in globally distributed software development," *IEEE Trans. Softw. Eng.*, vol. 29, no. 6, pp. 481–494, Jun. 2003.

[37] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE Softw.*, vol. 18, no. 2, pp. 16–20, Mar./Apr. 2001.

[38] P. J. Hinds and D. E. Bailey, "Out of sight, out of sync: Understanding conflict in distributed teams," *Organ. Sci.*, vol. 14, no. 6, pp. 615–632, 2003.

[39] J. H. Iversen, L. Mathiassen, and P. A. Nielsen, "Managing risk in software process improvement: An action research approach," *MIS Quart.*, vol. 28, no. 3, pp. 395–433, 2004.

[40] J. Jacobs, J. van Moll, P. Krause, R. Kusters, J. Trienekens, and A. Brombacher, "Exploring defect causes in products developed by virtual teams," *Inf. Softw. Technol.*, vol. 47, no. 6, pp. 399–410, 2005.

[41] S. L. Jarvenpaa and D. E. Leidner, "Communication and trust in global virtual teams," *Organ. Sci.*, vol. 10, no. 6, pp. 791–815, 1999.

[42] S. L. Jarvenpaa, T. R. Shaw, and D. S. Staples, "Toward contextualized theories of trust: The role of trust in global virtual teams," *Inf. Syst. Res.*, vol. 15, no. 3, pp. 250–267, 2004.

[43] J. L. Johnson and P. M. Podsakoff, "Journal influence in the field of management—An analysis using Salancik index in a dependency network," *Acad. Manage. J.*, vol. 37, no. 5, pp. 1392–1407, 1994.

[44] P. Kanawattanachai and Y. Yoo, "Dynamic nature of trust in virtual teams," *J. Strateg. Inf. Syst.*, vol. 11, no. 3/4, pp. 187–213, 2002.

[45] P. Katerattanakul, B. Han, and S. G. Hong, "Objective quality ranking of computing journals," *Commun. ACM*, vol. 46, no. 10, pp. 111–114, 2003.

[46] T. Kayworth and D. Leidner, "The global virtual manager: A prescription for success," *Eur. Manage. J.*, vol. 18, no. 2, pp. 183–194, 2000.

[47] T. R. Kayworth and D. E. Leidner, "Leadership effectiveness in global virtual teams," *J. Manage. Inf. Syst.*, vol. 18, no. 3, pp. 7–40, 2001.

[48] P. G. W. Keen and S. Scott-Morton, *Decision Support Systems: An Organizational Perspective*. Reading, MA: Addison-Wesley, 1978.

[49] M. Keil, "Pulling the plug: Software project management and the problem of project escalation," *MIS Quart.*, vol. 19, no. 4, pp. 421–447, 1995.

[50] M. Keil, P. E. Cule, K. Lyytinen, and R. C. Schmidt, "A framework for identifying software project risks," *Commun. ACM*, vol. 41, no. 11, pp. 76–83, 1998.

[51] A. M. Khalfan, "Information security considerations in IS/IT outsourcing projects: A descriptive case study of two sectors," *Int. J. Inf. Manage.*, vol. 24, no. 1, pp. 29–42, 2004.

[52] B. L. Kirkman, B. Rosen, P. E. Tesluk, and C. B. Gibson, "The impact of team empowerment on virtual team performance: The moderating role of face-to-face interaction," *Acad. Manage. J.*, vol. 47, no. 2, pp. 175–192, 2004.

[53] L. J. Kirsch, "Portfolios of control modes and IS project management," *Inf. Syst. Res.*, vol. 8, no. 3, pp. 215–239, 1997.

[54] L. J. Kirsch, "Deploying common systems globally: The dynamics of control," *Inf. Syst. Res.*, vol. 15, no. 4, pp. 374–395, 2004.

[55] R. Klepper, "The management of partnering development in I/S outsourcing," *J. Inf. Technol.*, vol. 10, no. 4, pp. 249–258, 1995.

[56] S. Krishna, S. Sahay, and G. Walsham, "Managing cross-cultural issues in global software outsourcing," *Commun. ACM*, vol. 47, no. 4, pp. 62–66, 2004.
[57] T. H. Kwon and R. Zmud, "Unifying the fragmented models of information systems implementation," in *Critical Issues in Information Systems Research*, Boland, Ed. Chichester, U.K.: Wiley, 1987, pp. 227–251.
[58] M. C. Lander, R. L. Purvis, G. E. McCray, and W. Leigh, "Trust-building mechanisms utilized in outsourced IS development projects: A case study," *Inf. Manage.*, vol. 41, no. 4, pp. 509–528, 2004.
[59] K. R. T. Larsen and C. R. McInerney, "Preparing to work in the virtual organization," *Inf. Manage.*, vol. 39, no. 6, pp. 445–456, 2002.
[60] H. J. Leavitt, "Applied organization change in industry: Structural, technical, and human approaches," in *Proc. New Perspect. Organ. Res.*, 1964, pp. 55–71.
[61] J. N. Lee and Y. G. Kim, "Understanding outsourcing partnership: A comparison of three theoretical perspectives," *IEEE Trans. Eng. Manag.*, vol. 52, no. 1, pp. 43–58, Feb. 2005.
[62] P. Lowry, D. Romans, and A. Curtis, "Global journal prestige and supporting disciplines: A scientometric study of information systems journals," *J. Assoc. Inf. Syst.*, vol. 5, no. 2, pp. 29–80, 2004.
[63] H. Lucas, *Implementation—The Key to Successful Information Systems*. New York: Wiley, 1981.
[64] J. S. Lurey and M. S. Raisinghani, "An empirical study of best practices in virtual teams," *Inf. Manage.*, vol. 38, no. 8, pp. 523–544, 2001.
[65] K. Lyytinen and R. Hirschheim, "Information systems failure—A survey and classification of the empirical literature," in *Oxford Surveys in Information Technology*. London, U.K.: Oxford Univ. Press, 1987, pp. 257–309.
[66] K. Lyytinen, L. Mathiassen, and J. Ropponen, "A framework for software risk management," *J. Inf. Technol.*, vol. 11, no. 4, pp. 275–285, 1996.
[67] K. Lyytinen, L. Mathiassen, and J. Ropponen, "Attention shaping and software risk—A categorical analysis of four classical risk management approaches," *Inf. Syst. Res.*, vol. 9, no. 3, pp. 233–255, 1998.
[68] A. Majchrzak, A. Malhotra, and R. John, "Perceived individual collaboration know-how development through information technology-enabled contextualization: Evidence from distributed teams," *Inf. Syst. Res.*, vol. 16, no. 1, pp. 9–27, 2005.
[69] A. Majchrzak, A. Malhotra, J. Stamps, and J. Lipnack, "Can absence make a team grow stronger," *Harv. Bus. Rev.*, vol. 82, no. 5, pp. 131–137, 2004.
[70] A. P. Massey, M. M. Montoya-Weiss, and Y. T. Hung, "Because time matters: Temporal coordination in global virtual project teams," *J. Manage. Inf. Syst.*, vol. 19, no. 4, pp. 129–155, 2003.
[71] L. Mathiassen and J. Stage, "The principle of limited reduction in software design," *Inf. Technol. People*, vol. 6, no. 2, pp. 171–185, 1992.
[72] L. Mathiassen, T. Saarinen, T. Tuunanen, and M. Rossi, "A contingency model for requirements development," *J. Assoc. Inf. Syst.*, vol. 8, no. 11, pp. 570–598, 2007.
[73] M. L. Maznevski and K. M. Chudoba, "Bridging space over time: Global virtual team dynamics and effectiveness," *Organ. Sci.*, vol. 11, no. 5, pp. 473–492, 2000.
[74] E. F. McDonough, K. B. Kahn, and A. Griffin, "Managing communication in global product development teams," *IEEE Trans. Eng. Manag.*, vol. 46, no. 4, pp. 375–386, Nov. 1999.
[75] F. W. McFarlan, "Portfolio approach to information systems," *Harv. Bus. Rev.*, vol. 59, no. 5, pp. 142–150, 1981.
[76] H. Minzberg, *Structure in Fives: Designing Effective Organizations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[77] M. M. Montoya-Weiss, A. P. Massey, and M. Song, "Getting it together: Temporal coordination and conflict management in global virtual teams," *Acad. Manage. J.*, vol. 44, no. 6, pp. 1251–1262, 2001.
[78] M. Mortensen and P. J. Hinds, "Conflict and shared identity in geographically distributed teams," *Int. J. Confl. Manage.*, vol. 12, no. 3, pp. 212–238, 2001.
[79] T. Moynihan, "An inventory of personal constructs for information systems project risk researchers," *J. Inf. Technol.*, vol. 11, no. 4, pp. 359–371, 1996.
[80] B. Nicholson and S. Sahay, "Some political and cultural issues in the globalisation of software development: Case experience from Britain and India," *Inf. Organ.*, vol. 11, no. 1, pp. 25–43, 2001.
[81] J. Nunamaker, M. Chen, and T. D. M. Purdin, "Systems development in information systems research," *J. Manage. Inf. Syst.*, vol. 7, no. 3, pp. 89–106, 1991.
[82] R. J. Ocker, "Influences on creativity in asynchronous virtual teams: A qualitative analysis of experimental teams," *IEEE Trans. Prof. Commun.*, vol. 48, no. 1, pp. 22–39, Mar. 2005.
[83] M. Ould, *Managing Software Quality and Business Risk*. Chichester, U.K.: Wiley, 1999.
[84] N. Panteli and R. M. Davison, "The role of subgroups in the communication patterns of global virtual teams," *IEEE Trans. Prof. Commun.*, vol. 48, no. 2, pp. 191–200, Jun. 2005.
[85] S. Paul, I. M. Samarah, P. Seetharaman, and P. P. Mykytyn, "An empirical investigation of collaborative conflict management style in group support system-based global virtual teams," *J. Manage. Inf. Syst.*, vol. 21, no. 3, pp. 185–222, 2004.
[86] S. Paul, P. Seetharaman, I. Samarah, and P. P. Mykytyn, "Impact of heterogeneity and collaborative conflict management style on the performance of synchronous global virtual teams," *Inf. Manage.*, vol. 41, no. 3, pp. 303–321, 2004.
[87] D. J. Pauleen, "An inductively derived model of leader-initiated relationship building with virtual team members," *J. Manage. Inf. Syst.*, vol. 20, no. 3, pp. 227–256, 2003.
[88] D. J. Pauleen and P. Yoong, "Relationship building and the use of ICT in boundary-crossing virtual teams: A facilitator's perspective," *J. Inf. Technol.*, vol. 16, no. 4, pp. 205–220, 2001.
[89] G. Piccoli and B. Ives, "Trust and the unintended effects of behavior control in virtual teams," *MIS Quart.*, vol. 27, no. 3, pp. 365–395, 2003.
[90] R. E. Potter and P. A. Balthazard, "Virtual team interaction styles: Assessment and effects," *Int. J. Hum. Comput. Stud.*, vol. 56, no. 4, pp. 423–443, 2002.
[91] A. Powell, G. Piccoli, and B. Ives, "Virtual teams: A review of current literature and directions for future research," *Database Adv. Inf. Syst.*, vol. 35, no. 1, pp. 6–36, 2004.
[92] R. K. Rainer and M. D. Miller, "Examining differences across journal rankings," *Commun. ACM*, vol. 48, no. 2, pp. 91–94, 2005.
[93] N. Ramarapu, M. J. Parzinger, and A. A. Lado, "Issues in foreign outsourcing—Focus on applications development and support," *Inf. Syst. Manage.*, vol. 14, no. 2, pp. 27–31, 1997.
[94] V. Ramesh and A. R. Dennis, "The object-oriented team: Lessons for virtual teams from global software development," presented at the 35th Annu. Hawaii Int. Conf. Syst. Sci., Hawaii, HI, 2002.
[95] J. Ropponen and K. Lyytinen, "Components of software development risk: How to address them? A project manager survey," *IEEE Trans. Softw. Eng.*, vol. 26, no. 2, pp. 98–112, Feb. 2000.
[96] J. Rubin, *Handbook of Usability Testing—How to Plan, Design and Conduct Effective Tests*. New York: Wiley, 1994.
[97] A. F. Rutkowski, D. R. Vogel, M. Van Genuchten, T. M. A. Bemelmans, and M. Favier, "E-collaboration: The reality of virtuality," *IEEE Trans. Prof. Commun.*, vol. 45, no. 4, pp. 219–230, Dec. 2002.
[98] R. Sabherwal, "The role of trust in outsourced IS development projects," *Commun. ACM*, vol. 42, no. 2, pp. 80–86, 1999.
[99] S. Sakthivel, "Virtual workgroups in offshore systems development," *Inf. Softw. Technol.*, vol. 47, no. 5, pp. 305–318, 2005.
[100] S. Sarker and S. Sahay, "Implications of space and time for distributed work: An interpretive study of US–Norwegian systems development teams," *Eur. J. Inf. Syst.*, vol. 13, no. 1, pp. 3–20, 2004.
[101] S. Sarker, S. Sarker, D. B. Nicholson, and K. D. Joshi, "Knowledge transfer in virtual systems development teams: An exploratory study of four key enablers," *IEEE Trans. Prof. Commun.*, vol. 48, no. 2, pp. 201–218, Jun. 2005.
[102] D. A. Schön, *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books, 1983.
[103] Y. Shin, "A person–environment fit model for virtual organizations," *J. Manage.*, vol. 30, no. 5, pp. 725–743, 2004.
[104] C. M. Solomon, "Global teams—The ultimate collaboration," *Pers. J.*, vol. 74, no. 9, pp. 49–58, 1995.
[105] B. C. Y. Tan, K. K. Wei, W. W. Huang, and G. N. Ng, "A dialogue technique to enhance electronic communication in virtual teams," *IEEE Trans. Prof. Commun.*, vol. 43, no. 2, pp. 153–165, Jun. 2000.
[106] H. Taylor, "Critical risks in outsourced IT projects: The intractable and the unforeseen," *Commun. ACM*, vol. 49, no. 11, pp. 75–79, 2006.
[107] (2005). Thomson Scientific, Web of Science Sort Options [Online]. Available: http://www.apps.isiknowledge.com/WoS//help/h_sort.htm#sorting
[108] A. M. Townsend, S. M. DeMarie, and A. R. Hendrickson, "Virtual teams: Technology and the workplace of the future," *Acad. Manage. Exec.*, vol. 12, no. 3, pp. 17–29, 1998.

[109] T. Van Der Smagt, "Enhancing virtual teams: Social relations v. communication technology," *Ind. Manage. Data Syst.*, vol. 100, no. 3/4, pp. 148–156, 2000.

[110] D. R. Vogel, M. van Genuchten, D. Lou, S. Verveen, M. van Eekout, and A. Adams, "Exploratory research on the role of national and professional cultures in a distributed learning project," *IEEE Trans. Prof. Commun.*, vol. 44, no. 2, pp. 114–125, Jun. 2001.

[111] M. Warkentin and P. M. Beranek, "Training to improve virtual team communication," *Inf. Syst. J.*, vol. 9, no. 4, pp. 271–289, 1999.

[112] M. E. Warkentin, L. Sayeed, and R. Hightower, "Virtual teams versus face-to-face teams: An exploratory study of a Web-based conference system," *Decis. Sci.*, vol. 28, no. 4, pp. 975–996, 1997.

[113] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS Quart.*, vol. 26, no. 2, pp. XIII–XXIII, 2002.

[114] P. Weill and M. H. Olson, "An assessment of the contingency theory of management information systems," *J. Manage. Inf. Syst.*, vol. 6, no. 1, pp. 59–85, 1989.

[115] M. E. Whitman, A. R. Hendrickson, and A. M. Townsend, "Research commentary. Academic rewards for teaching, research, and service: Data and discourse," *Inf. Syst. Res.*, vol. 10, no. 2, pp. 99–109, 1999.

[116] Y. J. Xue, C. S. Sankar, and V. W. A. Mbarika, "Information technology outsourcing and virtual team," *J. Comput. Inf. Syst.*, vol. 45, no. 2, pp. 9–16, 2004.

[117] I. Zigurs, "Leadership in virtual teams: Oxymoron or opportunity?," *Organ. Dyn.*, vol. 31, no. 4, pp. 339–351, 2003.

John Stouby Persson received the Master's degree in informatics from Aalborg University, Aalborg, Denmark, in 2006. He is currently working toward the Ph.D. degree at the Department of Computer Science, Aalborg University.

His current research interests include information systems and software engineering with a particular emphasis on management of distributed software projects.

Mr. Persson is a member of the Association for Information System (AIS).

Lars Mathiassen (M'90) received the Master's degree in computer science from Aarhus University, Arhus, Denmark, in 1975, the Ph.D. degree in informatics from Oslo University, Oslo, Norway, in 1981, and the Dr. Techn. degree in software engineering from Aalborg University, Aalborg, Denmark, in 1998.

He is currently a Georgia Research Alliance (GRA) Eminent Scholar and a Professor in Department of Computer Information Systems at Georgia State University, Atlanta, where he is a Cofounder of center for process innovation. His current research interests include information systems and software engineering with a particular emphasis on process innovation. He has coauthored several books like, *Computers in Context* (Blackwell, 1993), *Object Oriented Analysis & Design* (Marko Publishing, 2000), and *Improving Software Organizations* (Addison-Wesley, 2002). He is the Senior Editor for the *MIS Quarterly* and his research is published in journals like *Information Systems Research, MIS Quarterly*, the IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, *Communications of the ACM, Journal of* Association for Information System (AIS), *Information Systems Journal*, and the IEEE SOFTWARE.

Dr. Mathiassen is a member of the Association for Computing Machinery (ACM) and the AIS.

Jesper Boeg received the Master's degree in informatics from Aalborg University, Aalborg, Denmark, in 2006.

He is currently working as a Project Manager at Trifork A/S, Aarhus, Denmark. He is dedicated to achieving maximum performance and efficiency through the use of the agile management framework Scrum with particular emphasis on self organizing teams, retrospectives and close customer collaboration. His current research interests include information systems and software engineering with a particular emphasis on process innovation, in recent years particularly in the area of agile system development.

Thomas Stenskrog Madsen received the Master's degree in informatics from Aalborg University, Aalborg, Denmark, in 2006.

He is currently employed by the Capability Maturity Model Integration (CMMI) level five company, Systematic A/S, Aarhus, Arhus, Denmark, where he is certified Scrum Master as well as user experience engineer. He is currently assigned to a joint distributed project with another Aarhus based company. His current research interests include agile and CMMI, and related project management.

Flemming Steinson received the Master's degree in informatics from Aalborg University, Aalborg, Denmark, in 2006.

He is currently employed by the Capability Maturity Model Integration (CMMI) level five company, Systematic A/S, Aarhus, Aarhus, Denmark, where he is risk manager as well as user experience engineer. His current research interests include process improvements and innovation in the area of user experience and interaction design.

Accepted to IEEE Software

A Process for Managing Risks in Distributed Teams

John S. Persson and Lars Mathiassen

Abstract

Distributed software projects represent particular risks that need careful managerial attention. We present a process for managing these risks that apply to distributed team structures. The process is based on state-of-the-art literature on distributed team risks, resolution techniques addressing these risks, and guidelines for how resolution techniques apply to risks. Following CMMI's generic approach to risk management, the process offers a series of rigorous steps that are readily understood and easy to follow. We illustrate the process based on a large-scale, strategic software project with multiple subprojects that cross organizational and national boundaries. In conclusion, we provide suggestions for tailoring the process to a particular project or organization.

Keywords: Distributed Teams, Risk Management, CMMI

Introduction

Today, many software projects are geographically distributed. Software mangers therefore need to know about distributed team management [8, 15]: how is team building done across sites; how is the task broken down and distributed; how is knowledge shared across time, space, and differences in culture; and, how is work coordinated to produce coherent outcomes? While the current literature offers rich insights into these challenges [5, 6, 12, 15], there is no process available for managing risks that apply to distributed team structures. On this backdrop, we present a comprehensive process compliant with CMMI. CMMI is a maturity model that organizations can use to assess and improve their processes. The model offers a comprehensive set of generic processes to support development of products and services [3]. One generic process focuses on risk management to help identify and analyze potential problems before they occur so that risk-handling activities can be planned and invoked across the project life-cycle. We adapt this generic approach to risk management [3] and describe a series of rigorous steps that are readily understood and easy to follow. In that way, we avoid redundancies and focus on the essentials of managing risk in distributed teams [10]. Practicing risk management is, however, challenging requiring nontrivial skills and insights [2]. So, while we provide a rigorous recipe for action, it is important to keep in mind that recipes always have to be adapted to become useful in practice:

> *A recipe tells you the ingredients, how to mix the ingredients, what temperature to use, and how long to cook those ingredients. However, it does not teach you the techniques of slicing and dicing, mixing, beating, whipping, blanching, grilling, poaching, etc. And recipes also leave room for some experimentation and modification.* [10]

Process Foundation

To create a solid foundation for the process, we synthesized state-of-the-art research on managing distributed teams [12]. Analyzing a total of 72 scientific articles, we identified the inherent risks in distributed teams, the resolution techniques proposed to address them, and the guidelines for how to apply resolution techniques to risks. Subsequently we integrated these findings into a web-based tool to support risk management in distributed teams. An implementation of the tool is freely available at www.distributedprojects.net.

We synthesized the risks presented in the literature into a two level taxonomy. First, we identified eight risk areas [12]: task distribution, knowledge management, geographical distribution, collaboration structure, cultural distribution, stakeholder relations, communication infrastructure, and technology setup. Second, we identified specific risk factors to further characterize each risk area [2]. Table 2 in the Appendix defines the identified risk areas and factors that are characteristic for distributed teams. While some of these might appear to be typical for any project the focus is here on risks related to distribution of teams. Distributed environments may significantly exacerbate each of the risks and traditional project management approaches may no longer be appropriate.

For example, one risk factor is collaboration capability under the risk area collaboration structure [8, 13, 14]. Collaboration capability describes team members' understanding and appreciation of differences in competencies and their ability to effectively use technology to gather and share information across geographical and functional distances [12, 13]. It is problematic for a distributed team if members have limited understanding of other members' competencies [13], for example in cases where a distributed team needs to manage software requirements across sites [1, 4, 8]. Another risk factor is language barriers under the risk area cultural distribution [8, 13, 14]. Language barriers arise when distributed team members do not share the same language or norms of communication. Such situations can easily lead to misinterpretations and un-conveyed information [12], both well-known problems in distributed teams [5, 6, 13, 15].

We also synthesized four different types of resolution techniques [12]: planning, control, social integration, and technical integration [11]. Planning techniques help design and organize projects so they can be effectively executed in distributed contexts; control techniques facilitate tracking progress across sites and help manage discrepancies in relation to plans; social integration techniques integrate team members and help manage cultural differences across sites; and, technical integration techniques increase connectivity and technical compatibility across sites. Table 1 offers a portfolio of specific techniques of each of these types that managers can adopt to mitigate distributed team risks. Finally, we identified guidelines for how to apply resolution techniques to distributed team risks [12], see Table 1. The eight areas of risks, the four types of resolution techniques, and the guidelines for combining them are hence syntheses of state-of-the-art research on distributed teams and they form the conceptual foundation for the presented risk management process.

Process Architecture

The risk management process is structured into three steps as illustrated in Figure 1: identify and analyze risks, develop risk mitigation plans, and implement risk mitigation plans. The process follows a risk-action list approach by offering directions for how to apply the four types of resolution techniques to the eight areas of distributed team risks [9]. As the focus is on risks related to or exacerbated by distribution of projects, managers need to adopt other risk management processes for complete risk management. Managers of distributed teams can prepare for risk management, the first practice goal for risk management in CMMI [3], by adopting the presented process.

The process also helps managers of distributed teams identify and analyze risks, the second practice goal for risk management in CMMI [3]. As risks are identified and categorized, team members evaluate risk probabilities and impacts and they prioritize how to address risks, cf. the first step in Figure 1. It is particularly important to involve team members during this step as the project manager may have limited knowledge about different sites. For a given project, the first level of analysis is therefore focused on developing risk estimates at each site. The second level of analysis focuses on developing risk estimates for the entire project based on the local estimates from the first step. This requires a collocated or mediated project meeting, in which participants uncover differences in perspectives and experiences across sites and negotiate how to prioritize overall distributed team risks. The resulting assessment can be distributed to the rest of the organization, allowing for risk management across sub-projects and comparisons and learning between independent projects.

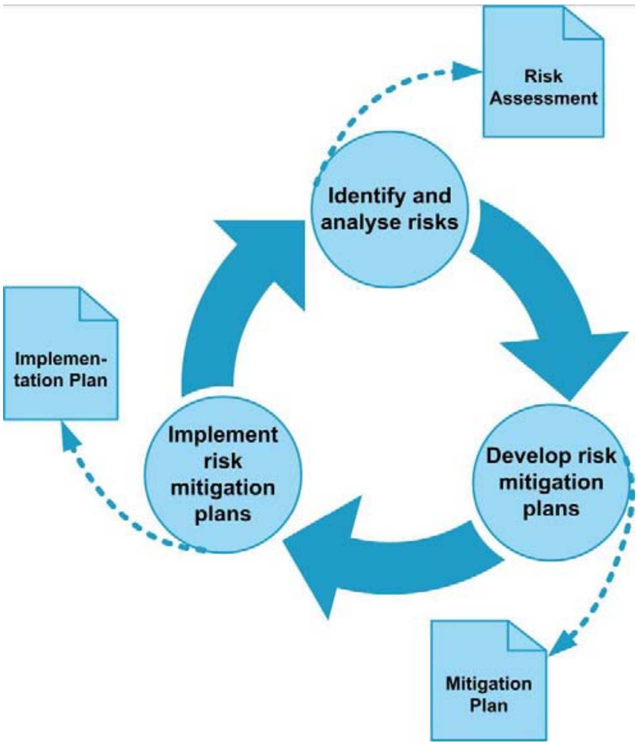

Figure 1 Steps and outcomes in the risk management process

The process also helps managers of distributed teams mitigate risks, the last goal for risk management in CMMI [3]. Following the CMMI process, mitigating risks include two specific practices, developing and implementing risk mitigation plans as illustrated in Figure 1. Development of a risk mitigation plan is supported by the list of resolution techniques and guidelines for how to apply these to address specific risk areas, cf. Table 1. During this step, participants can adopt resolution techniques from the list or develop novel resolution techniques to address distributed team risks in their project. The final step in the process is implementation of risk mitigation plans. In this step, participants relate to project objectives and decide on practical approaches. To do so, they consider responsibilities, resources, deliverables, and milestones as key elements in implementing risk mitigation plans [2].

Process Illustration

We illustrate the risk management process with a software project from *ScandicBank*[1], a large financial company based in northern Europe. *ScandicBank* has a long history of national mergers and is now expanding through acquisitions of companies in neighboring countries. Each acquisition requires a significant effort from *ScandicBank*'s IT-division. The strategy is to achieve economies of scale by implementing the bank's standard IT-platform as quickly as possible in all new branches. The responsibility for the IT-platform resides in *ScandicBank*'s head quarters. However, some acquired companies have their own IT-departments, and these are typically engaged in making the IT-platform adhere to specific requirements for financial software systems in their respective countries. The most recent acquisition by *ScandicBank* is a company rather different

[1] The illustration is based on data from a real-world distributed software project. We use a pseudonym to protect the identity of the firm.

from previous acquisitions. It is significantly larger, it has an existing sophisticated IT-platform, and it is located in a country with a language tradition very different from the dominant language within *ScandicBank*. Previous acquisitions were smaller, had an inferior IT-platform, and a language tradition similar to *ScandicBank*.

The implementation of the IT-platform is organized as *ScandicBank's Integration Project* (IP) illustrated in Figure 2. The project consists of 20 sub-projects, it has more than five hundred team members over its life-cycle, and it has a strict one year deadline. The team members are located at the IT-department of the acquired company and at four different sites of *ScandicBank*'s IT-division. Furthermore, as IP requires an unusual high number of IT-professionals, an Indian software outsourcing provider is engaged. Seven of the 20 sub-projects have team members distributed across all three physical locations, as illustrated in Figure 2; eleven sub-projects are distributed across *ScandicBank*'s IT-division and the IT-department of the acquisition or the Indian outsourcing provider; and only two sub-projects are not distributed.

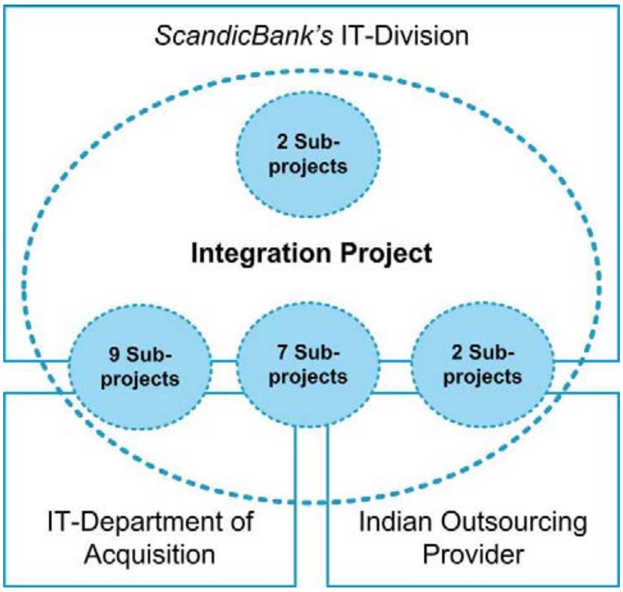

Figure 2 *ScandicBank*'s Integration Project

Identify and Analyze Risks

Identification and analysis of risks is a nontrivial task in distributed teams. There are many different types of risks involved and the knowledge needed to uncover them is typically distributed across team members and sites. Identification and analysis of risks is supported by Table 2 in the Appendix in which the eight risk areas and related risks factors are defined by analytical questions and evaluation criteria related to a three point scale (low, medium, high). The two-step approach takes into account the difficulties involved in communicating and sharing knowledge across distributed teams. In the first step, participants from each site identify and analyze risks based on Table 2. In the next step, the local assessments are discussed at joint meeting across sites to arrive at an overall risk assessment. This assessment seeks consensus on how to prioritize risks across sites. In large projects consisting of multiple sub-projects, each sub-project assessment would finally form the basis for a risk management meeting between sub-project representatives.

Table 2 in the Appendix is applied as follows to identify and analyze distributed team risks. For each risk factor, an assessment is made of the probability of unsatisfactory outcome P(UO) and the loss to the parties affected if the outcome is unsatisfactory L(UO) [2]. These assessments can for example be made on a scale with the numeric values 0 to 8, categorized into low (0-2), medium (3-

Accepted to IEEE Software

5), and high (6-8). The probability assessment is supported by the qualitative indicators of low, medium, and high risk in the table. Next, for each risk factor the risk exposure (RE) is calculated based on the equation RE = P(UO) * L(UO) [2]. The average RE for the three risk factors subsequently constitutes the risk area's RE value. Based on these values, a prioritized list of the eight risk areas are derived representing the significant risk areas.

In *ScandicBank*'s IT-division, the challenges related to geographical distribution of software teams is given high attention. Limited experience with distributed teams and the strategic importance of the most recent acquisition, prompt management to adopt the risk management process. The web-based tool [12] is made available in the company's software methodology portfolio and the process is included in the IT-divisions process library. The IP manager initiates the process early during the requirement phase. Each sub-project manager engages team members from the involved sites to do a local risk assessment based on Table 2 in the Appendix and as preparation to a joint risk management meeting. At this point, the IP manager also decides to repeat the risk management process with frequent intervals throughout the project's life-cycle.

Several sub-projects are in the fortunate position to have their first risk management meetings face-to-face, since co-located collaboration is emphasized for a few weeks during the requirements phase. Multi-national requirements engineering is thus predominantly carried out in a co-located setting. This early stage co-location is feasible because there are only few team members during the requirements phase (100), compared to later in the project (up to 500). Local risk assessments of each of the risk factors are presented and discussed using the web-based tool and a projector.

Some sub-projects are not able to meet face-to–face. They conduct the risk management meetings using teleconferencing and computer desktop sharing of the results from the web-based tool. In the beginning, participants are reluctant to comment on each others' risk assessments. In an effort to kick start the discussion, one project manager points out the details of her risk assessment and volunteers on what grounds they were given. To create an open and safe communication climate [5], she then directly asks one of her more experienced and outgoing colleagues, not afraid to disagree with her, to elaborate on his assessment. Similar initiatives can be taken to overcome differences between high context culture (India) and low context culture (Northern Europe) views in distributed teams [7].

With all sub-project risk assessments in hand, the IP project manager calls for a risk management meeting with all sub-project managers. The procedure for this meeting is similar to that of the sub-projects. However, the participants do not compare and contrast local assessments from each site, but instead the assessments of each sub-project. In the first meeting during the requirements phase, cultural distribution is identified as the most significant risk area across sub-projects. Also, it becomes clear some sub-projects have low assessments of all risk areas. These sub-projects are primarily located at a single site within the IT-division of *ScandicBank* with only a marginal number of team members at the IT-department of the acquisitioned company. As a consequence, it is decided only to include 6 out of 20 sub-projects in future risk management, see Figure 2. At later stages of the IP, task distribution is evaluated as the most significant risk area. Over the project's life-cycle, task distribution and cultural distribution frequently had high risk exposure assessments while communication infrastructure and geographical distribution frequently had low risk exposure assessments.

Develop Risk Mitigation Plans

Awareness of high priority risk areas is an important first step in addressing them. However, it is not sufficient. Effective risk management is based on comprehensive risk mitigation plans. A wide variety of resolution techniques is available for the development of risk mitigation plans, see Table 1. Each resolution technique applies to different risk areas and each risk area can be mitigated by different resolution techniques. In some situations the proposed resolution techniques are in opposition to each other, such as "*standardize and train in methods across sites*" on one hand, and "*handle differences in methods between sites*" on the other. The challenge is therefore to select a portfolio of coherent techniques that effectively addresses the risks at hand. Instead of adopting the suggested generic resolution techniques, participants can develop novel or company specific resolution techniques to address prioritized risks in their project. The development of risk mitigation plans take place at the joint risk management meeting based on the overall risk assessments for the entire project.

Planning	**Task Distribution**	**Knowledge Management**	**Geographical Distribution**	**Collaboration Structure**	**Cultural Distribution**	**Stakeholder Relations**	**Communication Infrastructure**	**Technology Setup**
Acquire complementary skills	X		X	X	X	X	X	X
Adjust meetings to distributed context	X	X	X				X	X
Divide tasks systematically between sites	X	X		X				
Reduce coupling between sites	X		X	X				
Create shared collaboration platform		X	X	X	X	X		X
Establish shared goals	X	X	X	X		X		
Establish communication norms		X	X		X	X	X	X
Define roles and responsibilities			X	X	X	X		
Reduce time-zone differences			X					
Control								
Focus on deliverables		X	X					X
Establish task coordination between sites	X			X				
Maintain site autonomy	X		X	X		X	X	
Establish shared control mechanisms	X	X	X	X		X		
Establish temporal coordination mechanisms	X	X	X	X				X
Maintain project organization overview	X	X	X	X			X	X
Maintain task overview within and across sites	X	X	X				X	X
Monitor and improve communication		X	X			X		X
Maintain a supportive environment	X		X		X	X		X
Analyze and manage errors	X	X						
Social Integration								
Improve capability to manage cultural differences		X	X		X			
Improve distributed collaboration skills			X	X		X	X	
Improve language skills					X			X
Emphasize early teambuilding activities			X			X	X	X
Promote humor and openness			X		X	X		
Use mentors to integrate new members	X	X				X	X	
Use face-to-face meetings appropriately	X	X	X	X		X	X	
Develop liaisons between sites	X	X	X		X	X	X	
Adopt shared reward systems				X	X	X		
Technical Integration								
Increase technical compatibility between sites		X	X				X	X
Standardize and train in methods across sites	X	X		X	X			X
Adopt appropriate communication technologies	X	X	X		X	X	X	X
Improve collaboration and communication technology skills		X	X	X		X	X	X
Improve development technology skills		X						X
Handle differences in methods between sites		X		X				
Combine waterfall model and prototyping	X			X				

Table 1 Develop risk mitigation plans

Accepted to IEEE Software

In the initial risk management of the IP, resolution techniques are adopted both at the project and sub-project level. One sub-project with cultural distribution as main priority, choose the resolution technique "*improve capability to manage cultural differences"*. In the web-based tool, detailed resolution suggestions are provided as follows [12]:

- *Establish courses in cultural diversity during the startup phase of the project. If team members are stationed at remote sites, the cultural training should take place before departure.*
- *Focus on creating understanding and acceptance of differences, e.g., by letting each team member make a presentation on their individual culture, values, and expectations.*
- *Promote understanding and acceptance rather than seek to streamline the project organization.*
- *Focus on the strengths that diversity offers rather than the weaknesses.*
- *Acknowledge and discuss cultural differences in a respectful and civilized manner, and keep in mind that there are limits to cultural adaptation.*
- *Adjust management style according to culture, e.g., team members' preferences for well-defined tasks vs. preference for loosely defined task and self-management.*

Not all resolution suggestions are practically or financially feasible in the sub-project. However, the specific resolution suggestions spur debate and more specific ideas from team members on how to address cultural distribution. The second resolution suggestion inspires a discussion resulting in an agreement of having a lunch meeting where one team member from each of the three countries would make a short presentation of their corporate and national culture. The participants furthermore discuss what strengths their cultural diversity could offer and personal preferences regarding well-defined versus loosely defined tasks.

Resolution techniques for cultural distribution are also discussed at a risk management meeting for the IP at large. The sub-project managers discuss which resolution techniques to initiate across sub-projects. It is agreed to primarily mitigate risks at this level by "*developing liaisons between sites"*, see Table 1. Each sub-project with more than 10 team members will have liaisons at least at one *ScandicBank* IT-division site, while sub-projects with more than 25 team members will have liaisons at multiple sites. Task distribution is discussed at a later risk management meeting. It is agreed to primarily mitigate risks through the technique "*standardize and train in methods across sites"*, see Table 1. Several sub-project managers argue that team members outside the *ScandicBank* IT-division lack knowledge of the development method, hindering effective task distribution among sites. The managers agree to organize development method training at the Indian site and the IT-department of the acquisition.

Implement Risk Mitigation Plans

In risk management, it is essential to reach conclusions leading to actions. The final step of the process is therefore developing implementation plans for each prioritized risk area, see Figure 3. These plans lay out the activities necessary to bring the related risk area under control. Each plan contains answers to five basic questions [2]: the objectives (why) are identified through the risks assessment; the deliverables and milestones (what and when) suggest when actions are to be taken; the responsibilities (who and where), describes which individuals are responsible for a given action, and where within the distributed organization it is to be carried out; the approach (how) consists of the identified resolution techniques; and, the resources (how much) estimate the costs associated with addressing the risk area. The mitigation plans should then be integrated into the project's overall risk management plan. In accordance with Figure 1, the risk management process should be repeated throughout the project life-cycle as risk profiles change [2, 3, 10]. Distributed teams should, therefore, decide how often to manage risks and make sure that each activation of the process is properly documented and reviewed in subsequent activations.

Accepted to IEEE Software

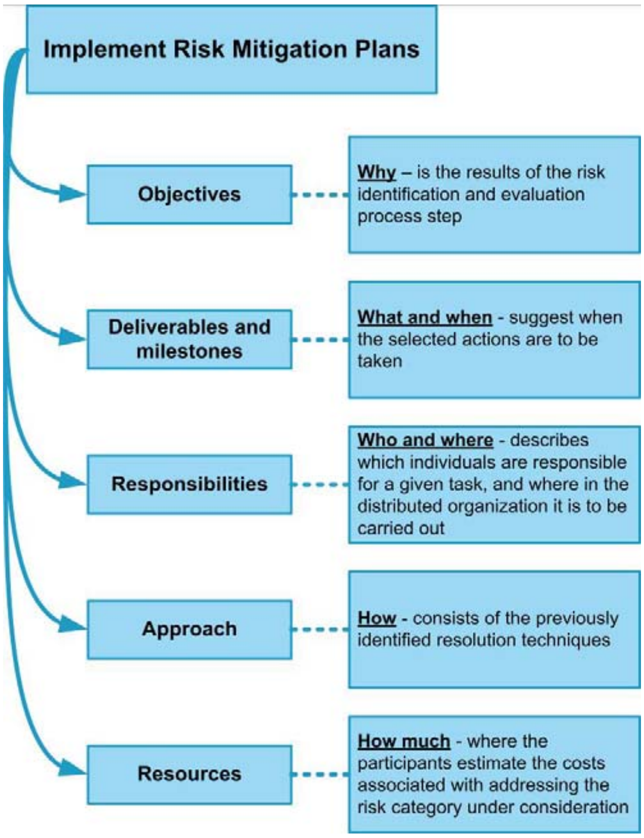

Figure 3 Implement risk mitigation plans

In the IP, the five elements of risk mitigation planning are detailed by the end of the initial joint risk management meetings. For example, in the before mentioned sub-project the plan to improve the ability to manage cultural differences is implemented as follows. The objective is to address cultural distribution as several team members expressed insecurity concerning how to contribute to the new distributed team. The deliverables and milestones in relation to cultural distribution risks are the cultural presentation meetings scheduled at specific dates. Concerning responsibilities, it is noted who would make presentations of their cultural background in each case. Adding to the approach, notes are made on what specific topics team members would be interested to learn more. Finally, it is estimated that the initiative would represent a monthly two hour resource load for each project. In the following revisit of the cultural distribution risk area, conducted 2 months later, risk exposures went from high to medium. Hence, the participants found their initiatives had reduced culturally related misunderstandings and eased collaboration between team members.

The risk management meeting for the overall IP also addresses the five elements to implement more off site training in *ScandicBank*'s development method. The objective is to address task distribution as participants complain about task equivocality, in particular related to unclear testing and documentation responsibilities across sites. The deliverable is an extension of *ScandicBank*'s method training program to be designed in three 3 days and delivered in two weeks. A participant from development support and a sub-project manager agrees to take responsibility for the training initiative. In discussing the approach, several sub-project managers request extended training in the areas of testing and documentation. These activities are estimated to represent sizable resources, primarily covered, however, by company-wide human resource efforts and thus not part of the IP budget. The time invested by team members is furthermore estimated to give significant productivity payoffs in the remainder of the project. In the revisit of the task distribution risk area 2 months later, risk exposure assessments were lower. Task equivocality was, however, still perceived as challenging. Therefore, in addition to the training initiative, co-location of team members was increased. While expensive, this additional initiative, which can be seen as an extreme adaptation of "develop liaisons between sites", proved successful in alleviating not only task distribution risks but also cultural distribution risks.

Process Tailoring

Generic processes can leave room for experimentation and offer suggestions on how to make modifications [10]. In tailoring the process for managing distributed team risks to match the preferences of your project or organization, you should therefore consider the following:

- *Keep it simple.* There are many challenges in managing distributed teams. It is important to establish a frequency of risk management that is appropriate for the organization and project. You don't want participants to see the process as an unrewarding administrative burden. In some cases the increased awareness resulting from risk management is more valuable than the resulting plans and initiatives.

- *Balance participation.* Distributed teams vary in size and complexity. In some cases, you only want to involve one or a few participants from each site. In other cases, you might find it useful to involve all team members. However, no matter the amount of participants, the establishment of an open and safe communication climate is vital for efficient risk management.

- *Adapt taxonomies.* The process is designed for a variety of preferences. Some elements of the architecture might not fit your project or organization. For example, you might want to include additional resolution techniques - a specific consultant, a particular course, available technologies - to the mix when developing risk mitigation plans, cf. Table 1.

- *Integrate plans.* Managing distributed team risks is only one aspect of risk management. Moreover, risk management is only one of many key disciplines in project management. For each project, you need to integrate the process into project management at large. Appropriate software systems can be a significant help in such integrations.

- *Attract capabilities.* Practicing the process will make it clear that specific capabilities are needed to effectively manage distributed teams. As you move forward through the project life-cycle, or as you engage in new distributed teams, you should attract complementary capabilities to your project to proactively reduce the probability and impact of distributed team risks.

Appendix

Risk Area	Risk Factor and Risk Question	Low Risk	Medium Risk	High Risk
Task Distribution	**Task Uncertainty** Do team members posses the knowledge and capabilities needed?	Team members know the task and it fits well with their capabilities.	Team members have reasonable task knowledge and their capabilities cover most challenges.	There are serious gaps in team members' task knowledge and required capabilities.
	Task Equivocality Do team members understand the specification of the task?	The task is well specified and understood by team members.	Most aspects of the specification are clear and the task is understood by key team members.	The specification lacks clarity on major points and many team members have limited task understanding.
	Task Coupling Is the task divided into distinct sub-tasks across sites?	There is minor need to coordinate development work across sites.	There is some need to coordinate development work across sites.	There is major need to coordinate development work across sites.

Knowledge Management	**Knowledge Creation** How is task knowledge created across sites?	All sites contribute well to creation of required task knowledge.	Most sites contribute reasonably well to creation of required task knowledge.	There are major problems related to creation of required task knowledge.
	Knowledge Capture How is task knowledge captured across sites?	Task knowledge is captured effectively across sites.	Task knowledge is with some exceptions captured effectively across sites.	There are major problems related to capturing task knowledge across sites.
	Knowledge Integration How is task knowledge integrated and shared across sites?	Task knowledge is integrated and shared well across sites.	Task knowledge is with some exceptions well integrated and shared across sites.	Task knowledge integration and sharing across sites is limited.
Geographical Distribution	**Spatial Distribution** How many sites are involved and what is the distance between them?	There are few sites collaborating over limited distance.	There are several sites collaborating over some distance.	There are many sites collaborating over considerable distance.
	Temporal Distribution How do time-zone differences impact development work?	Time-zone differences cause no or only minor problems.	Time-zone differences require some ad-hoc coordination across sites.	Time-zone differences cause major problems and require constant attention across sites.
	Goal Distribution How diverse are goals across sites?	Team members share major goals across sites.	There is some variation in goals across sites.	There are major goal conflicts across sites.
Collaboration Structure	**Collaboration Capability** Can team members collaborate across sites?	Team members collaborate across sites as needed.	In most cases, team members collaborate across sites as needed.	Breakdowns in collaboration across sites are common.
	Coordination Mechanisms Are coordination mechanisms appropriate across sites?	Coordination mechanisms are shared across sites and well adapted to the distributed context.	Coordination mechanisms are shared by most team members and reasonably well adapted to the distributed context.	Coordination mechanisms are not shared across sites and poorly adapted to the distributed context.
	Process Alignment Are processes aligned across sites?	Processes (including methods, templates, and guidelines) are shared across sites.	Processes (including methods, templates, and guidelines) vary, but are reasonably well aligned across sites.	Processes (including methods, templates, and guidelines) are different across sites.
Cultural Distribution	**Language Barriers** Do language and communication norms vary across sites?	Team members share language and communication norms across sites.	Team members use a common language with minor differences in communication norms.	Team members do not share a common language and have different communication norms.
	Work Culture Does work culture differ between sites?	Team members share work culture (including authority and team behavior) across sites.	Team members understand variations in work culture (including authority and team behavior) across sites.	Team members do not understand variations in work culture (including authority and team behavior) across sites.
	Cultural Bias Does cultural bias impact communication and cooperation across sites?	There are no major variations in cultural values across sites.	Team members communicate and collaborate based on appreciation of cultural variations across sites.	Team members lack knowledge of variations in cultural values across sites.
Stakeholder Relations	**Stakeholder Commitment** Are stakeholders committed to the project?	Key stakeholders are committed and share a common project identity across sites.	Most stakeholders are committed to the project and know about its distributed organization.	Stakeholder commitment varies and there is a lack of shared project identity.
	Mutual Trust Is there trust between stakeholders across sites?	There is appropriate mutual trust across sites.	There are instances of insufficient trust across sites.	Stakeholders do not trust each other across sites.
	Relationship Building Can the project integrate stakeholders across sites?	Existing and new stakeholders are well integrated across sites.	Existing and new stakeholders are mostly integrated well across sites.	There are several cases of stakeholders not being well integrated.
Communication Infrastructure	**Personal Communication** What is the level of personal communication and social interaction across sites?	The level of personal communication and social interaction across sites is appropriate.	There is some personal communication and social interaction across sites.	Personal communication and social interaction across sites is limited.
	Interaction Media How is communication across sites supported by media?	Communication needs across sites are well supported by media.	Communication across sites is for many purposes well supported by media.	There are severe shortcomings in media support of communication across sites.

	Teleconference Management How is teleconferencing managed across sites?	Teleconferencing is used appropriately and managed effectively across sites.	Teleconferencing is used to some extend across sites and reasonably well managed.	There is limited use of teleconferencing across sites and they are not managed well.
Technology Setup	**Network Capability** Are communication networks reliable?	Networks are not causing delays in development work and communication.	Networks are causing some delays in development work and communication.	Networks are causing serious delays in development work and communication.
	Tool Compatibility Are tools compatible across sites?	There are no compatibility issues between tools across sites.	Compatibility issues between tools create some collaboration barriers across sites.	Compatibility issues between tools create serious collaboration barriers across sites.
	Configuration Management How are configurations managed across sites?	There is appropriate configuration management across sites.	Configuration management is with some exceptions appropriate across sites.	There is limited configuration management across sites.

Table 2 Identify and analyze risks

References

[1] J.M. Bhat, M. Gupta, and S.N. Murthy, "Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing," *IEEE Software*, vol. 23, no. 5, 2006, pp. 38-44.

[2] B.W. Boehm, "Software Risk Management - Principles and Practices," *IEEE Software*, vol. 8, no. 1, 1991, pp. 32-41.

[3] CMMI-Product-Team, CMMI for Development, Version 1.2. 2006, Carnegie Mellon Software Engineering Institute: Pittsburg, PA.

[4] D. Damian, "Stakeholders in Global Requirements Engineering: Lessons Learned from Practice," *IEEE Software*, vol. 24, no. 2, 2007, pp. 21-27.

[5] C.B. Gibson and J.L. Gibbs, "Unpacking the Concept of Virtuality: The Effects of Geographic Dispersion, Electronic Dependence, Dynamic Structure, and National Diversity on Team Innovation," *Administrative Science Quarterly*, vol. 51, no. 3, 2006, pp. 451-495.

[6] D.C. Gumm, "Distribution Dimensions in Software Development Projects: A Taxonomy," *IEEE Software*, vol. 23, no. 5, 2006, pp. 45-51.

[7] E.T. Hall, *Beyond Culture*, New York: Anchor Press - Doubleday, 1977.

[8] J.D. Herbsleb and D. Moitra, "Global Software Development," *IEEE Software*, vol. 18, no. 2, 2001, pp. 16-20.

[9] J.H. Iversen, L. Mathiassen, and P.A. Nielsen, "Managing Risk in Software Process Improvement: An Action Research Approach," *MIS Quarterly*, vol. 28, no. 3, 2004, pp. 395-433.

[10] M.K. Kulpa and K.A. Johnson, *Interpreting the CMMI: A Process Improvement Approach*, Boca Raton, Florida: CRC Press, 2003.

[11] F.W. McFarlan, "Portfolio Approach to Information Systems," *Harvard Business Review*, vol. 59, no. 5, 1981, pp. 142-150.

[12] J.S. Persson, L. Mathiassen, J. Boeg, T.S. Madsen, and F. Steinson, "Managing Risks in Distributed Software Projects: An Integrative Framework," *IEEE transactions on Engineering Management*, vol. 56, no. 3, 2009.

[13] A. Powell, G. Piccoli, and B. Ives, "Virtual Teams: A Review of Current Literature and Directions for Future Research," *The DATA BASE for Advances in Information Systems*, vol. 35, no. 1, 2004, pp. 6-36.

[14] S. Sakthivel, "Virtual Workgroups in Offshore Systems Development," *Information and Software Technology*, vol. 47, no. 5, 2005, pp. 305-318.

[15] S. Sakthivel, "Managing Risks In Offshore Systems Development," *Communications of the ACM*, vol. 50, no. 4, 2007, pp. 69-75.

Submitted to Information Systems Journal

Enacting Control through Media and Context in Agile Distributed Software Development

John S. Persson, Lars Mathiassen, and Ivan Aaen

Abstract

While face-to-face interaction is fundamental in agile software development, distributed environments must rely extensively on mediated interactions. Practicing agile principles in distributed environments therefore pose particular challenges related to balancing fixed versus evolving quality requirements, and people versus process based collaboration. We report from an in-depth case study of a successful agile distributed software project with participants from a Russian and a Danish firm. Applying Kirsch's (2004) elements of control framework, we offer an analysis of how control was enacted through the project context and in the participants' mediated communication. Contrary to previous research, the analysis reveals two important insights. First, informal control mechanisms were used even though the team was short-lived and rarely met face-to-face. Second, informal roles and relationships such as clan-like control inherent in agile development was pervasively observed in this distributed setting. We discuss these insights and their implications in relation to theory and practice.

Keywords: Distributed development, agile project management, control theory.

Introduction

Two significant trends have emerged in software development practice over the past years: Agile methodologies and geographical distribution. These are recently being combined, so agile development methods are used in geographically distributed contexts (Agerfalk & Fitzgerald, 2006; Armour, 2007; Holmström et al., 2006; Pries-Heje et al., 2005; Paasivaara et al., 2008; Sutherland et al., 2007). Combining distributed development with agile principles does, however, represent particular control challenges related to balancing fixed versus evolving quality requirements, and people versus process based collaboration (Ramesh et al., 2006). While agile principles with lightweight processes and primary reliance on skilled people offer advantages in terms of flexibility, speed, and learning, there is a risk that agile practices in distributed environments further reinforce the well known difficulties related to collaboration and control of quality.

Studies of control in geographically distributed contexts have been conducted in relation to issues such as culture (Narayanaswamy & Henry, 2005), effectiveness (Piccoli & Ives, 2000), and trust (Gallivan, 2001; Gallivan & Depledge, 2003; Piccoli & Ives, 2003). Kirsch (2004), however, calls for research to more closely examine the role of the global context on control choices and impacts. The global context is described in terms of priority differences among locales, time zone, cultural variations, and geographical differences. Further calls for research on control in geographically distributed contexts has also been made by Powell et al. (2004). In their literature review, they question whether informal control mechanisms in general can be used when teams are short-lived and rarely meet face-to-face. This concern increases as agile principles are adopted because these principles mainly rely on face-to-face communication.

We investigate how different elements of control are enacted in successful agile distributed development; in particular how different elements of control are enacted through media usage and the project context. The following in-depth single case study investigates control enacted in media usage and context to manage a successful agile distributed development project. The investigated project spans geographical, cultural, and organizational boundaries between a Russian and a Danish firm. The project was successful by introducing a new product on the market which attracted investment from a Fortune500 Company. We thereby seek to contribute to the call for empirical research of agile software development (Dybå & Dingsøyr, 2008) in distributed settings (Agerfalk & Fitzgerald, 2006; Lee et al., 2006) in relation to control (Kirsch, 2004; Powell et al., 2004) by addressing the following research question:

How do successful agile distributed software projects enact different elements of control through mediated communication and project context?

Distributed Software Projects

Organizing project teams in a geographically distributed setting - frequently conceptualized as virtual teams - has received significant research attention (Hertel et al., 2005; Martins et al., 2004; Powell et al., 2004; Schiller & Mandviwalla, 2007). An important underlying theme running through this research is that virtual teamwork is characterized by challenges and paradoxes often hindering innovation (Dubé & Robey, 2009; Gibson & Gibbs, 2006). As a result, distributed software projects experience numerous management challenges (Iacovou & Nakatsu, 2008; Sakthivel, 2007). These challenges include unclear task coupling (Sakthivel, 2005; Sarker & Sahay, 2004), poor process alignment (Herbsleb & Moitra, 2001), differing work cultures (Dubé & Paré, 2001; Krishna et al., 2004), and inhibited communication capabilities (Bjørn & Ngwenyama, 2009; Herbsleb & Mockus, 2003). A wide range of alleviating initiatives have been suggested, such as, information and communication technologies (Bradner et al., 2005; Jang et al., 2002), risk management approaches (Persson et al., 2009; Sakthivel, 2007), communication training (Warkentin & Beranek, 1999), best practice lists (Battin et al., 2001; Ebert & De Neve, 2001), and social integration strategies (Kotlarsky & Oshri, 2005; van der Smagt, 2000). Among these initiatives, no silver bullet has been identified (Berry, 2008; Brooks, 1987) and new initiatives are continuously being studied.

Agile distributed development is a relatively new approach addressing some of the challenges in distributed software projects (Holmström et al., 2006; Pries-Heje et al., 2005). However, in line with the software engineering silver bullet conundrum, new solutions induce new challenges (Berry, 2008). Agile distributed software projects require particular attention to controlling the process and quality across sites (Ramesh et al., 2006). The agile characteristics of lightweight process, ongoing negotiation, and reliance on skilled people therefore induce new challenges related to balancing people versus process oriented control and fixed versus evolving quality requirements (Ramesh et al., 2006). On this backdrop, there is a need for increased research attention to understand control enactment in the context of agile distributed software teams. In particular, research is needed to understand the conditions under which informal control is used in these settings (Yadav et al., 2009).

Control of Software Projects

In the research of software development management, Kirsch (1996; 1997; 2000; 2004) has advocated the control perspective based on work by Ouchi (1978; 1979; 1980) and Eisenhardt (1985). Control is used broadly to denote any attempt to motivate individuals to behave in a manner consistent with organizational objectives (Kirsch, 2004; Ouchi, 1979). Control is viewed as either formal or informal. Formal control is a performance evaluation strategy, where either behaviors or outcomes are measured, evaluated, and rewarded (Eisenhardt, 1985; Kirsch, 1996). Informal control differs from formal control in that it is based on social and people strategies (Eisenhardt, 1985; Kirsch, 1996). Various modes or mechanisms of control are not applied separately (Ouchi, 1979).

Instead, they are part of portfolios of control mechanisms to support management practice in different organizational contexts (Kirsch, 1997).

Kirsch (2004) focuses on formal and informal control in relation to four elements of control. These four elements are: measurement; evaluation; rewards and sanctions; and roles and relationships, as shown in Table 1. The first three elements are based on a literature review by Eisenhardt (1985). The last element, roles and relationships, is an elaboration of the other basic elements added by Kirsch (2004).

Element	Formal	Informal
Measurement	• pre-specified and formally documented goals and/or behavior are available • control modes align the goals of controller and controllee • goals and/or behaviors are measurable	• few specified behaviors or procedures available • implicit specification and measurement of group values and norms • goals evolve over time • desired end-states result when individual behavior is consistent with the shared norms and values
Evaluation	• information about rules, procedures, behaviors, and goals are exchanged • information is exchanged in formal, written documents such as standard operating procedures or status reports • evaluation assesses whether behavior is resulting in forward progress	• information about norms, values and expectations exchanged • socialization, training, discussions, dialogs, and meetings serve as mechanisms of information exchange • goal of evaluation is to build and foster collegial relationships characterized by common values and norms
Rewards and sanctions	• based on following specified rules or achieving specified targets • formal organizational mechanisms include pay, bonuses, promotion or demotion	• based on acting in a manner that is consistent with group norms and values • mechanisms include group recognition and peer pressure
Roles and relationships	• focus is usually on dyads • controller and controllee are often in a formal superior-subordinate relationship or in a relationship that is consistent with the organizational hierarchy	• often a work group or professional society • may be a clan, which is a group of individuals who are dependent on each other to accomplish their work and who are committed to achieving group goals

Table 1 The elements of control (Kirsch, 2004)

Formal **measurement** implies behaviors or outcomes are explicitly specified and measurable, whereas informal measurement implies norms, values, or behavior are implicitly specified and measured. **Evaluation** refers to performance and information exchange. Formal evaluation is based on specified information regarding behavior and outcome, and assesses if current status leads to forward progress. Informal evaluation refers to norms and values that characterize a functional relationship assumed to lead to performance. The functional relationship is achieved by socialization through dialog or discussions. **Rewards and sanctions** are in a formal setting based on achieving specific goals or adhering to pre-specified behavior. Formal rewards could be bonuses, and formal sanctions could be demotions. Informal rewards and sanctions are based on whether a behavior is consistent with group values and norms. Informal rewards could be peer recognition, while informal sanctions could be social exclusion. **Roles and relationships** are added by Kirsch (2004) as an elaboration of the other three elements of control. Formal roles and relationships imply particular roles and usually a focus on dyadic relationships. Informal roles and relationships appear in groups of individuals dependent on each other and committed to group goals (Kirsch, 2004).

Research Approach

The research question was investigated through a single case study following the guidelines proposed by Yin (2003). The case study approach is well suited when the boundaries between a phenomenon and its context are unclear (Benbasat et al., 1987; Yin, 2003), as is the case in this study between control enactment and agile distributed contexts. A case study's capacity for investigating operational links (Yin, 2003) is also needed when pursuing explanatory knowledge, reflected in the use of a "how" research question. The single case study approach is furthermore suitable for testing the boundaries of well-formed theory (Benbasat et al., 1987; Yin, 2003), which is done by applying the control theory framework to understand the new practice of agile distributed development. Finally, a single case study is appropriate when the case is rare (Yin, 2003) as in this study, where we had rare access to video as well as audio recordings of mediated interactions in a successful distributed agile development project crossing organizational and cultural boundaries.

The case project was conceived by a small Danish software firm, *Area9*, as a joint venture with a Russian R&D outsourcing provider to finalize the development of a mindmapping tool to support management of distributed teams called *Comapping*. *Area9* was established in 2006 by two medical doctors and two computer scientists, all of whom previously had management positions in another software firm also relying heavily on offshore developers. *Area9* has had offshore developers in Russia and India from the very beginning. *Area9* base their development practices on agile principles, and manage distributed projects as if the developers where located in Denmark. This management practice requires that developers are able to work in an agile environment and have excellent communication and collaboration skills. *Area9*'s agile practices include among others continuous integration, parallel development and testing, incremental design, code reviews, and sparse documentation.

The case study data was collected over a half year period. The primary data was observations of how control was enacted through media usage to support e-conferences between the project participants. These observations were supplemented by a series of interviews with individual project participants, investigating how control was enacted through the project context in which the e-conferences took place. The communication media used in the e-conferences were Skype (www.skype.com) and a working prototype of the *Comapping* tool under development (www.comapping.com). The tool uses mindmapping diagrams to represent and share words, ideas, tasks, or other items linked to a central key word or idea in a project. The participants can work on such mindmaps in real time by visualizing the movements of each participant's cursor and inputs to existing or new nodes, similar to Google docs (docs.google.com). Even though the mindmapping tool had not been released for open beta when the case study was initiated, it had all its basic functionality. From the very start of the project, beta-versions of the tool were used for project coordination and hence played a vital role in the management of its own development. During the e-conferences the main author was present offsite as a passive observer while audio recording conversations and video recording activities in *Comapping*. The conference language was English and all e-conferences took place within normal working hours due to a time-zone difference of only two hours between the sites.

A total of ten observations and eleven semi-structured interviews were conducted, see Table 2. The people interviewed were; the manager (Denmark), who is board member of the *Comapping* project and director of Technology & Innovation at *Area9*; the developer (Russia), who is director of R&D for the *Comapping* project; the second developer (Russia), who is software developer in the *Comapping* project; the marketer (Denmark), who is CEO for the *Comapping* joint-venture; and the chairman (Denmark) of the joint-venture and CEO of *Area9*. The interviews were initiated with a face-to-face meeting with the manager. After this first meeting, observations were initiated and a series of interviews with key fulltime project participants were conducted through Skype. The interview objective was to investigate the emphasis on measurement, evaluation, rewards and sanctions, and roles and relations in the project context and the underlying reasoning behind the choices made. Towards the end of the case study, a new series of interviews were conducted with the project participants and the chairman, see Table 2. The objective of these interviews was to uncover participants' assessments of how control was enacted and possibly changed over the considered time-period through retrospective reasoning. During these interviews the *Comapping* tool was used by both the interviewer and interviewee to model a timeline of the project.

Date (2007)	Duration	Data Type	Actor(s)
January 19	71 min	Interview	Manager
February 26	42 min	Observation	Manager, Developer
Marts 6	77 min	Interview	Manager
Marts 22	26 min	Observation	Manager, Developer
	32 min	Interview	Developer
Marts 27	21 min	Observation	Manager, Developer
April 5	32 min	Interview	Second Developer
April 24	32 min	Observation	Marketer, Manager, Developer
	12 min	Interview	Manager
May 3	37 min	Observation	Marketer, Manager, Developer
May 8	56 min	Interview	Marketer
May 16	17 min	Observation	Marketer, Manager, Developer
May 21	64 min	Observation	Marketer, Manager, Developer
June 4	32 min	Observation	Marketer, Manager, Developer
June 26	50 min	Observation	Marketer, Manager, Developer
June 27	93 min	Interview	Manager
June 28	40 min	Interview	Marketer
July 2	15 min	Observation	Marketer, Manager, Developer
July 13	26 min	Interview	Developer
July 17	43 min	Interview	Second Developer
August 8	31 min	Interview	Chairman

Table 2 Interviews and observations

The data analysis was conducted with the qualitative data analysis software Atlas.ti V5.2 (Muhr, 2008). All fourteen hours of recordings were loaded into the software and carefully listened through. Statements in the interviews and exchanges in the form of one or more coherent decisions, claims, directives or mindmap activities during the e-conferences pertaining to control (see Table 1) were identified and coded as one of the four elements of control. In addition, we added to the coding descriptions of how the element of control under consideration was applied. In summary, the analysis of the interviews identified 94 statements related to how control was enacted through the project context, see Table 3, and the analysis of the e-conferences identified 54 exchanges related to how control was enacted through media communication, see Table 4.

Analysis

The *Comapping* project started at *Area9* February 2006 with an idea of a web-based mindmapping system. The idea was presented for their Russian outsourcing provider and the joint venture project between the two firms was established in April 2006. The two firms had equal ownership but made different contributions to the project. The Russian partner assigned two developers to the project while *Area9* provided management, architectural, and design expertise. Both of the IT professionals in *Area9* worked as full time developers on the project along with the two Russian developers, and together they had a proof of concept ready the following month. However, as the project evolved, the *Area9* participants decreased their contribution as developers, leaving more responsibility to the Russian developers. One of the *Area9* participants stopped as developer while the other who previously had the primary responsibility for product development handed over that role to one of the Russian developers. Another change was the hiring of three part time managers for the project in April 2006. The three managers were supposed to develop a commercial strategy for the product. However, as they spent their time debating options and challenges without being able to agree on a strategy, they were released from the project in December 2006. The chairman (Denmark) resumed management responsibilities along with the previous manager (Denmark) and managing developer (Russia).

We initiated this case study at this point in late February 2007. However, the new technically oriented management approach was short-lived. There was agreement that the project needed a business strategy for the product and one of the three managers therefore rejoined the project as full time CEO in April. A few months later a golden opportunity appeared as an Fortune500 Company partner was found. The new partner's customization became the primary project objective and staff was increased to eight full time developers. Our case study ended in July 2007 when the golden opportunity reorientation started.

Control was exercised in the project by multiple actors and in various forms. Table 3 summarizes the distribution of 94 statements about how control was enacted through the project context from the interviews. Table 4 summarizes the distribution of how control was enacted through mediated communication during the *Comapping* project e-conferences in 53 exchanges. The elements of control column show the number of identifications for each of the four control elements. In addition, the formal and informal columns show the number of control characteristics describing the identified instances of the elements of control. A particular element of control identification may have more than one control characteristic, e.g. instances of measurement can include both formal goal specification and documentation. Findings for each of the three basic elements of control (measurement, evaluation, and rewards and sanctions) are presented in the following with additional insights related to the fourth element (roles and relationships).

Elements of Control	Formal	Informal
Measurement (48)	Documented (9) Specified goals (12) Specified behavior (7) Goal alignment (5)	Norms and values (13) Evolving goals (14)
Evaluation (36)	Rules Procedures (3) Specified goals (3) Specified behavior (2) Documented (13)	Norms and values (12) Expectations (5) Socialization (5) Training (2) Dialogs (9)
Rewards and sanctions (10)	Rules Specified targets (1) Pay (4) Bonuses Promotion (1) Demotion (1)	Norms and values (5) Group recognition (5) Peer pressure (1)
Roles and relationships (94)	Dyad (12) Hierarchal (5)	Work group (43) Professional society (1) Clan (34)

Table 3 Control enacted through the project context based on interviews

Elements of Control	Formal	Informal
Measurement (24)	Documented (15) Specified goals (13) Specified behavior Goal alignment	Norms and values (1) Evolving goals (12)
Evaluation (24)	Rules Procedures (3) Specified goals (13) Specified behavior Documented (14)	Norms and values (1) Expectations (6) Socialization Training Dialogs (8)
Rewards and sanctions (5)	Rules Specified targets Pay Bonuses Promotion Demotion	Norms and values (1) Group recognition (3) Peer pressure (2)
Roles and relationships (53)	Dyad (5) Hierarchal	Work group (38) Professional society Clan (10)

Table 4 Control enacted through mediated communication based on observation of e-conferences

Measurement

The *Comapping* project context included both formal and informal measurement. In fact, measurement accounted for more than half of the statements relating to control, see Table 3. A majority of these were formal, however both formal and informal measurement was frequently based on informal roles and relationships. The chairman emphasized this reliance on informal roles and relationships in their measurement as fundamental from the very beginning of the *Comapping* project. According to him, the initial intent with the *Comapping* project was to "*first get the right people on the buss and then decide where to drive*". Based on past collaborations, there was a high level of trust between the managers of *Area9* and the Russian outsourcing firm in line with the importance of control and trust dialectics in inter-organizational electronic partnerships (Gallivan & Depledge, 2003). In fact, only a two page contract was needed before the joint venture project was established. The idea was to develop a web based collaborative mindmapping system *Area9* could use in other projects. The chairman's focus on getting the right people is fundamental in agile development (Augustine et al., 2005; Boehm & Turner, 2003) and illustrates a high reliance on informal measurement where "*desired end-states result when individual behavior is consistent with the shared norms and values*" (Kirsch, 2004). However, the use of a contract shows formal measurement was also present - although limited - because of the strong reliance on informal roles and relationships. The extensive reliance on the "right people", on high mutual dependency, and on trust indicates clan-like roles and relationships.

Control enactment through real-time mediation during e-conferences also included both formal and informal measurement. These e-conferences included three project members: the manager (board member of the *Comapping* joint-venture and responsible for general management and product development), the marketer (CEO for the *Comapping* joint-venture and responsible for its commercial strategy), and the developer (director of R&D for the *Comapping* joint-venture and responsible for day-to-day product development). During the e-conferences they shared and manipulated a mindmap (a tree structure) where each node represents an assignment to team members or groups to be done in a present ("Current sprint") or future ("Next sprint") time-box, a common technique for task management in agile development methodologies (Boehm & Turner, 2003; Jalote et al., 2004). As each participant could navigate the mindmap, all cursors were visible to the participants as a small box with the name of its owner, see Figure 1. Manipulations of the mindmap, e.g. deleting, adding, or changing a node, were instantly visible to all participants. Formal control during these e-conferences appeared frequently when participants specified goals in the mindmap. One example occurred as a result of the participants discussing a system feature represented as the node "Add notes to topic", under "Next sprint", Figure 1.

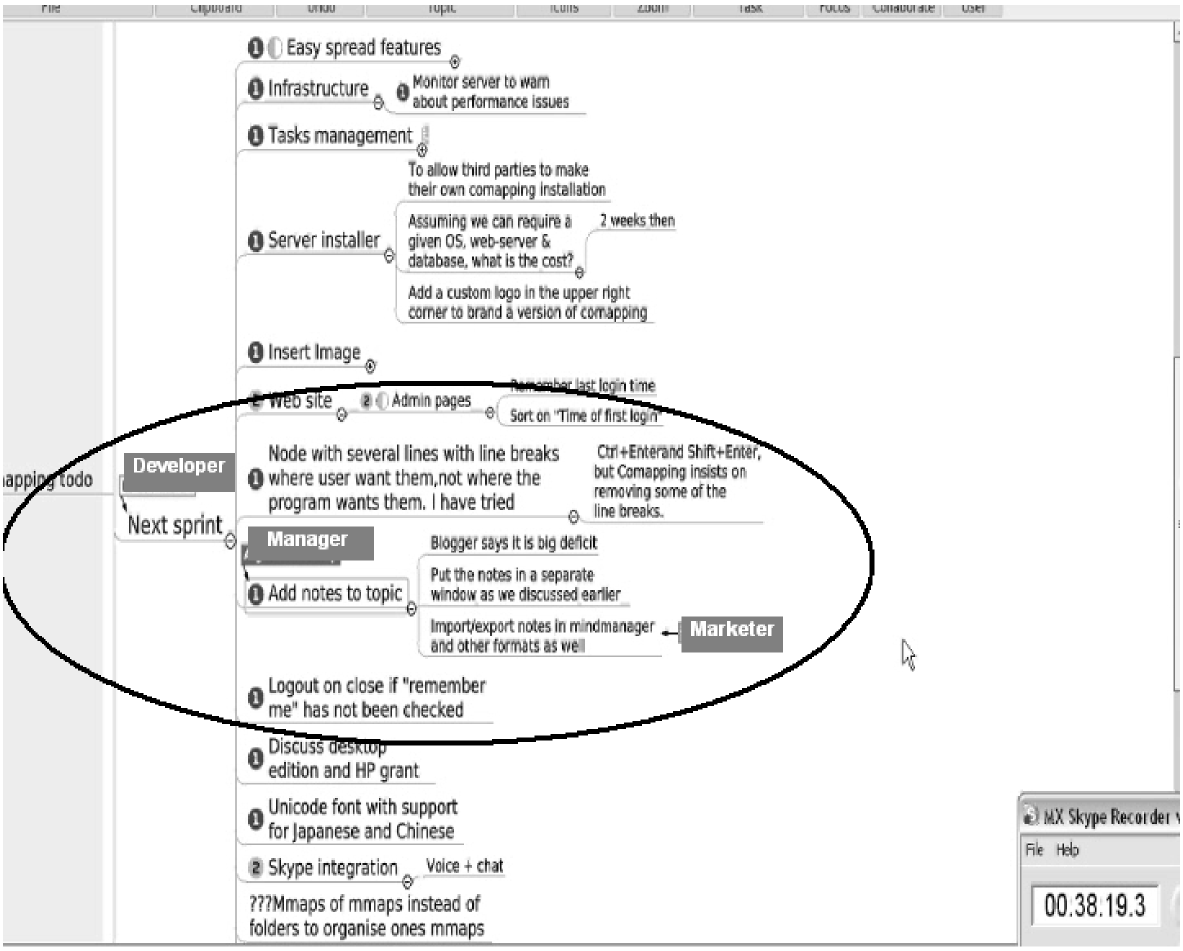

Figure 1 Screenshot of "Next sprint" in the *Comapping* tool as participants discuss the node "Add notes to topic"

In this case, the participants agreed that “Add notes to topic” needs to be done, resulting in the manager moving the node and its sub-nodes from “Next sprint” (Figure 1) to “Current sprint” (Figure 2). In addition, the manager stated:

Manager *(Denmark): I think it would be good to have the formatting also like we have on topics.*

Developer *(Russia): Formatting... okay... Maybe formatting can be second turn.*

Manager *(Denmark): Yeah it could, but we should definitely think about it.*

[The Manager creates a sub-node to “Add notes to topic,” named “Add formatting,” and marks it as second priority, see Figure 2]

Developer *(Russia): Okay, let me just.*

Manager *(Denmark): I added that as a second priority.*

This quote along with the screenshots is evidence of formal measurement through goal specification and documentation in the mindmap. However, informal roles and relationships are also illustrated in the third statement in the manager’s use of “we”.

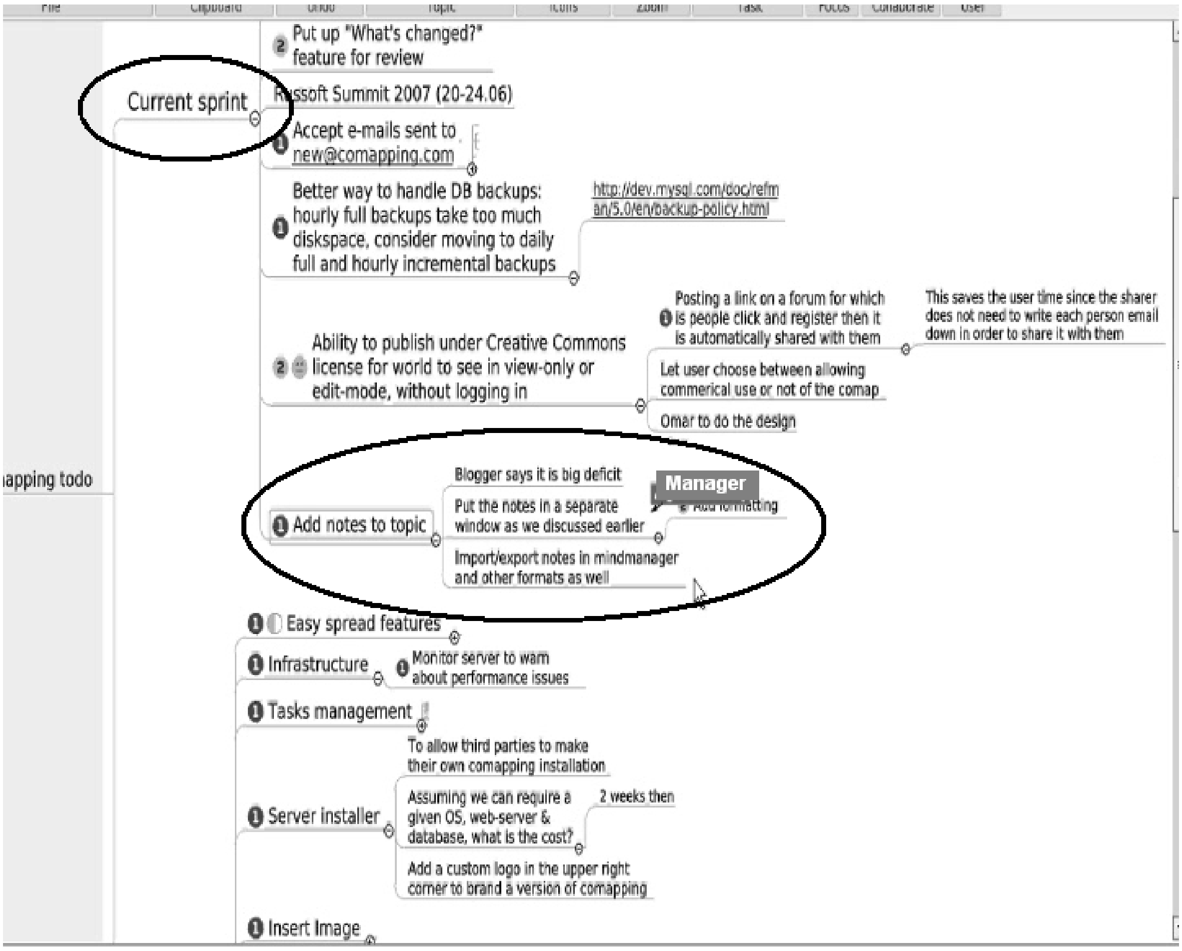

Figure 2 Screenshot of “Current sprint” in the *Comapping* tool as the manager has created “Add formatting” under “Put the notes in a separate window as we discussed earlier”

The specific goals of the project were, however, very dynamic and continuously negotiated. Goal changes often occurred by the end of e-conferences during discussions of the content and deadlines of the current sprint. A deadline was not always set for a sprint and in other instances deadlines were moved. The marketer tried several times to introduce more specific and stable goals. At one instance, he suggested a contingency plan of when to do server upgrade, but this proposal met resistance from the manager. He argued instead for an in-flight sense-and-response approach to deal with project dynamics. Differences of opinions between these two participants also appeared in relation to product features. The marketer had told people a desktop version of the system would be available within two months and he still wanted new features added to the current web-based version. The manager heatedly opposed this and argued that they needed to prioritize:

> [All the participants' cursors are placed on the sub-node of "Next sprint" called "Discuss desktop edition and HP grant" that however shortly after is changed to "Contact Xxx to hear about HP grant". At the time of the following quote only the marketer's cursor is placed on the node while the two others' cursors are placed on "Next sprint". This constellation of cursors was persistent for almost the entire discussion related to this particular node.]
>
> ***Marketer** (Denmark): I was asking if we can devote one resource so something is happening to the desktop version.*
>
> ***Manager** (Denmark): But that doesn't make sense, you need to answer the question, what is it you want from a business point of view, so if you say I want all of it you didn't answer the question.*
>
> ***Developer** (Russia): You have to say what you want first.*
>
> ***Manager** (Denmark): Yes, so it doesn't make sense to say we just need to keep the kettle burning on the desktop version, and therefore we should assign someone. That is not a good way to approach it I think. It is better to say what are the priorities, what is it I want to have first. Do you want the server installer before you want the desktop version or do you want the desktop version before you want the server installer or what is it you want?*
>
> ***Marketer** (Denmark) interrupts: Okay, then I'll suggest we will just wait with this for a while until we are more certain.*
>
> [The long discussion on this topic did not lead to further changes to this mindmap node]

The different placement of cursors in the mindmap in this exchange represents non verbal communication between the participants. The manager and developer's placement of cursors away from the node being discussed communicates an alliance between them that suggests it's time to move away from the topic or end the discussion. During this exchange, the developer and manager relied on the clan structure to convince the marketer that a high level of uncertainty requires the project to focus on a few prioritized features at a time. In fact, the manager stated in an interview that sprints and time-boxing were introduced precisely because the project earlier had experienced difficulties in prioritizing features and consequential low productivity.

Evaluation

Evaluation was both formal and informal in the *Comapping* project. The participants, however, differed significantly in their attention to evaluation through the project context. The marketer had a limited focus on evaluation in contrast to the manager and the two developers. The marketer illustrated this by his limited requests for information about activities in Russia. He said "*freedom with responsibility*", and did not expect his time was well spent on the technical aspects of the project. Instead, he focused on the marketing challenges. In addition, the marketer stated it would not make sense for him to provide feedback to the developers, because the joint venture was small and he trusted they worked as hard as they could. In contrast, the manager emphasized the importance of frequent contact with the remote site, arguing that if there were no exchanges for a week it was an indicator little work had been done. In addition, he stated that demands for communication should come from both sites, even though this could prove difficult. Hence, the manager emphasized frequent communication between him and the developers as the most crucial element in their agile practices, and he saw this capability as based on long term relationships and trust. According to the key developer, the manager and he chatted on a daily basis with very few exceptions, and used Skype on a weekly basis. Face-to-face communication was done on a monthly basis for the first 4-5 months of the *Comapping* project, but the developer considered the involvement of many new actors as the primary reason for these visits. While the developer preferred face-to-face communication, he did not consider it necessary between him and the manager. For the other developer, it varied significantly how frequently she communicated with the manager, from daily to weeks or months, depending on the specific task she was working on. In general, the manager found it difficult to understand how distributed software projects could be effective without being agile. He argued that plan driven development with extensive reliance on documents and limited verbal communication would go against their experience that misunderstandings frequently occur when reliance on written communication is high and verbal communication is low.

Evaluation through mediated communication during e-conferences was also both formal and informal. Formal controls were in the form of procedures, specified goals, and documentation. Informal controls were in the form of norms and values, expectations, and dialogs. An example of a frequent formal procedure was the review, which was agreed upon in the following quote:

[The developer and manager's cursors are placed on the first sub node of current sprint called "Easy sign up" marked as a first priority]

Manager *(Denmark): So do I need to do anything on that one [Easy sign up feature].*

Developer *(Russia): Probably not.*

Manager *(Denmark): So when you have something I can review it.*

Developer *(Russia): Yeah.*

Manager *(Denmark): Ok great, so it is fair to say it is 50% done.*

[The manager places the 50% symbol next to the node "Easy sign up feature"]

Developer *(Russia): Yes because it took time to understand how Joomla can be done.*

Review agreements were continuously made throughout e-conferences. It was not only the project participants doing reviews for each other; sometimes they planned reviews to be conducted by other people in their respective firms. In some cases, the participants also conducted reviews of new features during e-conferences by switching away from the *Comapping* tool to consider the current version of the system. There was also a documentation aspect of these reviews as the participants either wrote down who should do the review or, as in the example above, noted the task status in the mindmap.

The informal part of the evaluations took place as continuous negotiations of expectation:

[The manager is relocating the sub nodes of "Current sprint" so they are ordered by their priority]

Marketer *(Denmark): Are you okay with this, or is it too much?*

Developer *(Russia): Let me just do a quick review – This should be fine, I think. The number ones are definitely doable shall we say two weeks as usual?*

Manager *(Denmark): If you prefer so that would be fine.*

Developer *(Russia): Okay let's go for two weeks – if we do well we can move faster – but I'm sure some other things will be popping up.*

[The manager creates a new sub node of "Current sprint" called "Deadline: 4th of June"]

In this exchange, the tree participants made an effort to balance expectations supported by the mindmap. The readily available representation of expectations in the mindmap provided the participants with a dynamically updated overview leading to a final agreement. The marketer checked the other participants were comfortable with the accumulated expectations by the end of the e-conference (start of the above quote) and the developer subsequently reminded the others of the usual uncertainties that might cause delays.

Rewards and Sanctions

The *Comapping* project participants had limited focus on rewards and sanctions in their accounts of how control was enacted through the context. Formal rewards relied on the key participants' ownership of the joint venture. The developer had a small ownership and was also paid by the Russian outsourcing firm, while the marketer had a large ownership and was not getting any salary. Due to the team's small size they did not exercise promotions or degradations, performance pay, or bonus systems. Considering informal rewards, the manager argued the project was prestigious with high involvement by the Russian outsourcing firm's top management and cutting edge technology. According to him, *Area9* also emphasized the importance of the product's actual qualities for the Russia developers. One developer stated accordingly the delivery of a useful product was her most important goal and the other developer compared to a previous project: "*I didn't care about that product at all ... what I care (in this project) is to make the customer happy in whatever way he wants and make my team happy*". Regarding sanctions, the manager pointed out they tried to avoid a "*name and blame culture*", though it could be difficult on critical issues. Possible sanctions included taking low-performing participants off the team, which had been the case for some part-time programmers. In addition, he emphasized inadequate performing members might impact team cohesion and morale by inducing further work or problems for other team members. The two developers rarely experienced articulation of positive feedback; they mainly relied on the absence of negative feedback as an indicator of acceptance.

Reward and sanction controls were rarely enacted during e-conferences. No formal controls were observed and only a few of an informal nature. One example of group recognition occurred as the manager commented on the work by the developer:

[This quote is from the beginning of the e-conference where neither of them is logged on to Comapping]

Manager *(Denmark): So you made the map work with Joomla.*

Developer *(Russia): Yes.*

Manager *(Denmark): It looks good.*

Developer *(Russia) after two seconds of silence: And I actually found out I had to patch Joomla a bit to accept emails as user names – that is already done.*

Manager *(Denmark): Great.*

In addition, the participants applied sanctions in the form of peer pressure. The following quote concerns the amount of features in the first update of the *Comapping* tool since its public release:

[All the participants' cursors are placed on the sub node of "Current sprint" called "Add notes to topic" marked as a first priority and as 75% done, next to the nodes "Publish map" and "Share to all" marked with same priority and completion]

***Marketer** (Denmark): So we just have [the features] publish maps and add notes?*

***Developer** (Russia): Yes, and the new share dialog…*

***Marketer** (Denmark): … It's been six weeks right - and we are going into the seventh week - so in two months we come out with two features.*

***Developer** (Russia): Which is not good I agree - Which is not impressive.*

***Marketer** (Denmark): No…*

In this exchange, the marketer indirectly pressures the developer to implement more features for the upcoming system update (The e-conference was held June 26th and the update was released July 11th including six new features). This particular peer pressure relied on an implicit norm of what is an acceptable level of productivity for a given period of time. Further, it was supported by the clan mentality focusing on "we", underlining mutual dependence and shared productivity norms.

Discussion

In the following, we review the detailed analysis of the *Comapping* project in relation to our research question: How do successful agile distributed software projects enact different elements of control through mediated communication and project context?

Previous research questions whether informal control mechanisms in general can be used when teams are short-lived and rarely meet face-to-face (Powell et al., 2004). Findings from the *Comapping* project show a team distributed across geography, culture, and firms having success with the product they developed, while being highly reliant on informal controls. In fact, we found evidence of all of Kirsch's (2004) elements of control in the participant statements about the project context and our own observations of technology mediated real-time exchanges during the project's e-conferences. Moreover, considerable elements of informal as well as formal control were enacted both through the context and media usage even though emphasis was uneven across the elements. Comparing the distributions of the 94 statements on the context (see Table 3) with the 53 exchanges in the e-conferences (Table 4), notable differences and similarities are present. While the informal elements of control are distributed similarly when comparing statements about the project context and mediated exchanges during e-conferences, there are significant differences regarding the formal elements of control. These differences are concerning formal rewards and sanctions which were not enacted at all during e-conferences and formal measurement and evaluation which were observed significantly more frequently during e-conferences compared to the project context. While this suggests that contextual controls were more informal than those enacted through real-time mediated communication, it should be noted that roles and relationships were more frequently informal during e-conference exchanges.

Informal control enactment during e-conferences can be due to meetings in general being mostly an informal exchange mechanism (Kirsch, 2004). On the other hand, it is for this reason surprising to observe the extensive formal measurement and evaluation controls during e-conferences. However, usage of the collaborative mindmapping tool facilitated formalized measurement and evaluation through documentation and goal specification, an option not readily available in ordinary face-to-face meetings. Compared to face-to-face communication, lean communication media such as teleconferencing may hence call for more structure in the form of procedures and documents because of the absence of traditional nonverbal visual cues. The application of the *Comapping* tool in the project appeared to address this need by providing alternative visual cues closely related to the projects unfolding task profile. The *Comapping* tool might in this way prove useful in other agile distributed software projects experiencing difficulties during mediated meetings.

Previous research suggests that informal roles and relationships such as the clan-like control inherent in agile development, will likely be more difficult to practice in a distributed setting (Harris et al., 2006). Harris et al. (2006) argue that clan control can be increasingly difficult when interaction is not face-to-face but mediated by technology, and when participants come from different organizations (e.g. when consultants are used or when development is partly outsourced). Contrary to this claim, the results from the *Comapping* project suggest that agile practices and clan-like controls were pervasively observable both in real-time mediated exchanges between project participants and in their accounts of the project context. While the participants did have hierarchical controls available (as indicated by their formal titles) in the project context, we found no instances in which formal, hierarchical relationships were expressed during e-conferences. The reliance on informal roles and relationships instead of formal titles during e-conferences, supports the standard agile principles of getting the right people and establishing mutual trust (Augustine et al., 2005; Boehm & Turner, 2003). The high number of informal work group statements and exchanges suggest consistent high attention to the task at hand, while the second most frequent clan-like pattern indicates a high level of mutual dependence among team members. The clan pattern is not surprising considering the agile principles adopted in the *Comapping* project. It is, however, surprising when considering the argued difficulties for clan control in distributed settings (Harris et al., 2006). The appearance and effectiveness of clan controls in the *Comapping* project can be explained in two ways: first, because the high level of trust and long term relationships between not only the participants, but also the two firms, lead to fewer formal and more informal controls (Kirsch, 2004); second, because of the *Comapping* project's high goal congruence and high performance ambiguity (Ouchi, 1980). The informal roles and relationships allowed all participants to act as both controller and controllee in their continuous attempts to influence decision making during e-conferences (Choudhury & Sabherwal, 2003), a situation different from the traditional distinction in software development between controllers exercising control and controllees delivering on the agreed tasks to meet desired objectives (Kirsch & Cummings, 1996; Kirsch et al., 2002).

Conclusion

The success of agile practices in collocated software development stipulates cautious consideration of its potential in distributed settings. We reported how elements of control can be enacted through mediated communication and project context to support success in distributed, agile software development. The considered project was controlled through formal as well as informal measurement, evaluation, and rewards and sanctions, and with very high dependency on informal roles and relationships. More specifically, the case revealed two important insights. First, informal control mechanisms were used even though the team was short-lived and rarely met face-to-face which is an achievement questioned possible by previous research (Powell et al., 2004). Second, informal roles and relationships such as clan-like control inherent in agile development were pervasively observed in this distributed setting, which also is a remarkable finding taking previous research (Harris et al., 2006) into account. These findings contribute to the calls for empirical research of agile software development (Dybå & Dingsøyr, 2008) in distributed settings (Agerfalk & Fitzgerald, 2006; Lee et al., 2006) in relation to control (Kirsch, 2004; Powell et al., 2004). More specifically, the study addresses the call for research to understand the conditions under which informal control is used in agile distributed software projects (Yadav et al., 2009). The findings also point in direction of further investigations to help us better understand how different forms of control can be enacted to support software development in distributed, agile environments. In particular, multiple case studies are needed to compare and contrast control patterns across different agile and distributed software projects. Important variables related to control enactment in such studies could be communication media, project size, and participant diversity across both successful and unsuccessful projects.

REFERENCES

Agerfalk, P.J., & Fitzgerald, B. (2006) Flexible and distributed software processes: Old petunias in new bowls? *Communications of the ACM*, **49**, 26-34.

Armour, P.G. (2007) Agile ... and offshore - An interview with a new paradigm. *Communications of the ACM*, **50**, 13-16.

Augustine, S., Payne, B., Sencindiver, F., & Woodcock, S. (2005) Agile Project Management: Steering from the edges. *Communications of the ACM*, **48**, 85-89.

Battin, R.D., Crocker, R., Kreidler, J., & Subramanian, K. (2001) Leveraging resources in global software development. *IEEE Software*, **18**, 70-77.

Benbasat, I., Goldstein, D.K., & Mead, M. (1987) The case research strategy in studies of information systems. *MIS Quarterly*, **11**, 369-386.

Berry, D.M. (2008) The Software Engineering Silver Bullet Conundrum. *IEEE Software*, **25**, 18-19.

Bjørn, P., & Ngwenyama, O. (2009) Virtual team collaboration: building shared meaning, resolving breakdowns and creating translucence. *Information Systems Journal*, **19**, 227-253.

Boehm, B.W., & Turner, R. (2003) *Balancing agility and discipline: A guide for the perplexed* Addison-Wesley Professional, Boston, MA.

Bradner, E., Mark, G., & Hertel, T.D. (2005) Team size and technology fit: Participation, awareness, and rapport in distributed teams. *IEEE Transactions on Professional Communication*, **48**, 68-77.

Brooks, F.P. (1987) No silver bullet: Essence and accidents of software engineering. *IEEE Computer*, **20**, 10-19.

Choudhury, V., & Sabherwal, R. (2003) Portfolios of control in outsourced software development projects. *Information Systems Research*, **14**, 291-314.

Dubé, L., & Paré, G. (2001) Global virtual teams. *Communications of the ACM*, **44**, 71-73.

Dubé, L., & Robey, D. (2009) Surviving the paradoxes of virtual teamwork. *Information Systems Journal*, **19**, 3-30.

Dybå, T., & Dingsøyr, T. (2008) Empirical studies of agile software development: A systematic review. *Information and Software Technology*, **50**, 833-859.

Ebert, C., & De Neve, P. (2001) Surviving global software development. *IEEE Software*, **18**, 62-69.

Eisenhardt, K.M. (1985) Control - Organizational and Economic Approaches. *Management Science*, **31**, 134-149.

Gallivan, M.J. (2001) Striking a balance between trust and control in a virtual organization: a content analysis of open source software case studies. *Information Systems Journal*, **11**, 277-304.

Gallivan, M.J., & Depledge, G. (2003) Trust, control and the role of interorganizational systems in electronic partnerships. *Information Systems Journal*, **13**, 159-190.

Gibson, C.B., & Gibbs, J.L. (2006) Unpacking the Concept of Virtuality: The Effects of Geographic Dispersion, Electronic Dependence, Dynamic Structure, and National Diversity on Team Innovation. *Administrative Science Quarterly*, **51**, 451-495.

Harris, M., Hevner, A.R., & Collins, R.W. (2006) Controls in Flexible Software Development. In: *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, *Hawaii*.

Herbsleb, J.D., & Mockus, A. (2003) An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, **29**, 481-494.

Herbsleb, J.D., & Moitra, D. (2001) Global software development. *IEEE Software*, **18**, 16-20.

Hertel, G., Geister, S., & Konradt, U. (2005) Managing Virtual Teams: A Review of Current Empirical Research. *Human Resource Management Review*, **15**, 69-95.

Holmström, H., Fitzgerald, B., Ågerfalk, P.J., & Conchúir, E.Ó. (2006) Agile Practices Reduce Distance in Global Software Development. *Information Systems Management*, **23**, 7-18.

Iacovou, C.L., & Nakatsu, R. (2008) A risk profile of offshore-outsourced development projects. *Communications of the ACM*, **51**, 89-94.

Jalote, P., Palit, A., Kurien, P., & Peethamber, V.T. (2004) Timeboxing: a process model for iterative software development. *Journal of Systems and Software*, **70**, 117-127.

Jang, C.-Y., Steinfield, C., & Pfaff, B. (2002) Virtual team awareness and groupware support: an evaluation of the TeamSCOPE system. *International Journal of Human-Computer Studies*, **56**, 109-126.

Kirsch, L.J. (1996) The management of complex tasks in organizations: Controlling the systems development process. *Organization Science*, **7**, 1-21.

Kirsch, L.J. (1997) Portfolios of control modes and IS project management. *Information Systems Research*, **8**, 215-239.

Kirsch, L.J. (2000) Software project management: An integrated perspective for an emerging paradigm. in: *Framing the Domain of IT Management: Projecting the Future Through the Past,* R.W. Zmud (ed.), PinnFlex Education Resources, Cincinnati, OH, 285–304.

Kirsch, L.J. (2004) Deploying common systems globally: The dynamics of control. *Information Systems Research*, **15**, 374-395.

Kirsch, L.J., & Cummings, L.L. (1996) Contextual influences on self-control of is professionals engaged in systems development. *Accounting, Management and Information Technologies*, **6**, 191-219.

Kirsch, L.J., Sambamurthy, V., Ko, D.G., & Purvis, R.L. (2002) Controlling information systems development projects: The view from the client. *Management Science*, **48**, 484-498.

Kotlarsky, J., & Oshri, I. (2005) Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems*, **14**, 37-48.

Krishna, S., Sahay, S., & Walsham, G. (2004) Managing cross-cultural issues in global software outsourcing. *Communications of the ACM*, **47**, 62-66.

Lee, O.K., Banerjee, P., Lim, K.H., Kumar, K., Van Hillegersberg, J., & Wei, K.K. (2006) Aligning it components to achieve agility in globally distributed system development. *Communications of the ACM*, **49**, 48-54.

Martins, L.L., Gilson, L.L., & Maynard, M.T. (2004) Virtual Teams: What Do We Know and Where Do We Go From Here? *Journal of Management*, **30**, 805-835.

Muhr, T. (2008) ATLAS.ti - The knowledge workbench, Scientific Software Development, Berlin, Germany.

Narayanaswamy, R., & Henry, R.M. (2005) Effects of culture on control mechanisms in offshore outsourced IT projects. In: *Proceedings of the 2005 ACM SIGMIS on Computer Personnel Research*, *Atlanta, Georgia, USA*, 139-145.

Ouchi, W.G. (1978) Transmission of Control through Organizational Hierarchy. *Academy of Management Journal*, **21**, 173-192.

Ouchi, W.G. (1979) Conceptual-Framework for the Design of Organizational Control Mechanisms. *Management Science*, **25**, 833-848.

Ouchi, W.G. (1980) Markets, Bureaucracies, and Clans. *Administrative Science Quarterly*, **25**, 129-141.

Persson, J.S., Mathiassen, L., Boeg, J., Madsen, T.S., & Steinson, F. (2009) Managing Risks in Distributed Software Projects: An Integrative Framework. *IEEE Transactions on Engineering Management*, **56**,

Piccoli, G., & Ives, B. (2000) Virtual teams: managerial behavior control's impact on team effectiveness. In: *Proceedings of the twenty first international conference on Information systems*, *Brisbane, Queensland, Australia*, 575-580.

Piccoli, G., & Ives, B. (2003) Trust and the unintended effects of behavior control in virtual teams. *MIS Quarterly*, **27**, 365-395.

Powell, A., Piccoli, G., & Ives, B. (2004) Virtual Teams: A Review of Current Literature and Directions for Future Research. *DATA BASE for Advances in Information Systems*, **35**, 6-36.

Pries-Heje, J., Baskerville, R., & Hansen, G.I. (2005) Strategy models for enabling offshore outsourcing: Russian short-cycle-time software development. *Information Technology for Development*, **11**, 5-30.

Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2008) Using scrum in a globally distributed project: a case study. *Software Process: Improvement and Practice*, **13**, 527-544.

Ramesh, B., Cao, L., Mohan, K., & Xu, P. (2006) Can distributed software development be agile? *Communications of the ACM*, **49**, 41-46.

Sakthivel, S. (2005) Virtual Workgroups in Offshore Systems Development. *Information and Software Technology*, **47**, 305-318.

Sakthivel, S. (2007) Managing Risks In Offshore Systems Development. *Communications of the ACM*, **50**, 69-75.

Sarker, S., & Sahay, S. (2004) Implications of Space and Time for Distributed Work: An Interpretive Study of US-Norwegian Systems Development Teams. *European Journal of Information Systems*, **13**, 3-20.

Schiller, S.Z., & Mandviwalla, M. (2007) Virtual Team Research: An Analysis of Theory Use and a Framework for Theory Appropriation. *Small Group Research*, **38**, 12-59.

Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007) Distributed scrum: Agile project management with outsourced development teams. In: *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*.

van der Smagt, T. (2000) Enhancing virtual teams: social relations v. communication technology. *Industrial Management & Data Systems*, **100**, 148-156.

Warkentin, M., & Beranek, P.M. (1999) Training to improve virtual team communication. *Information Systems Journal*, **9**, 271-289.

Yadav, V., Adya, M., Sridhar, V., & Nath, D. (2009) Flexible global software development: antecedents of success in requirements analysis. *Journal of Global Information Management,*, **17**, 1-31.

Yin, R.K. (2003) *Case Study Research: Design and Methods*, (Third ed.) Sage Publications Inc, Thousand Oaks, California.

Submitted to Information Systems Research

Collective Minding in Virtual Teams:

Investigating Heedful Interrelating with Multimodal Communication

John S. Persson and Lars Mathiassen

Abstract

Virtual teams face significant coordination challenges, yet the literature is silent on how multimodal communication can help address these challenges. Against this backdrop, we present a framework that draws on Weick and Robert's (1993) collective mind notion to investigate how multimodal communication affects virtual team coordination. We use the framework to analyze the patterns of mediated interaction in a virtual team that coordinated its efforts using a combination of teleconferencing and real-time collaborative modeling. The analyses are based on audio recordings of the team's oral exchanges together with video recordings of its collaborative modeling efforts. As a result, we present empirical evidence that teleconferencing and real-time collaborative modeling facilitated collective minding and contributed to successful outcomes in the considered project; we offer theoretical propositions that explain how multimodal communication can facilitate collective minding and positively affect coordination performance in virtual teams; and we argue that the constructs of articulation acts, manipulation acts, and communication breakdowns provide a useful basis for investigating heedful interrelating in virtual teams based on multimodal communication.

Keywords: Virtual team, collective mind, coordination, multimodal communication

Introduction

The Internet and associated technologies have made it easy to communicate in real time across the globe through new communication channels, such as instant messaging, net meetings, and video conferences. These technologies allow team operations to become increasingly virtual (Bergiel et al. 2008). However, virtual teams still experience significant and persistent challenges in coordinating mediated by information technologies (Gibson and Gibbs 2006, Powell et al. 2004).

The ability to coordinate effectively in organizations has been conceptualized as a socio-cognitive phenomenon through the notion of a collective mind (Weick and Roberts 1993). The collective mind has proven useful in researching coordination performance in virtual teams (Yoo and Kanawattanachai 2001) and studies have called for such examinations of virtual team interaction over time and space (Fiore et al. 2003). The literature is, however, silent on whether it is possible to establish a collective mind in virtual teams based on mediated communication. While much of the current knowledge about virtual teams suggests it might be difficult (Anderson et al. 2007), this literature does not consider the impact of synchronous and multimodal communication on virtual team coordination. In addition, while macro and processual research approaches are available for investigating the collective mind in co-located contexts (Cooren 2004, Weick and Roberts 1993), no research approaches are currently available for investigating the collective mind based on multimodal communication.

Based on access to unusual and rich data from a successful virtual team, we were able to study detailed patterns of mediated, multimodal communication that led to high coordination performance. Coordination between two sites and across management, marketing, and product development was achieved based on a combination of teleconferencing and real-time collaborative modeling. This allowed us to analyze the collective mind and coordination based on a combination of audio recordings of the team's oral exchanges and video recordings of its real-time collaborative modeling. In addition, we had access to data from interviews with key stakeholders, archival data, and research notes from site visits. These unusual case characteristics allowed us to go beyond the typical analyses in virtual team research, where data consist of transcripts of lean media communications, participants' accounts gathered through interviews or questionnaires, and archival data. Instead of mainly emphasizing individual perceptions of team coordination (Anderson et al. 2007, Cooren 2004), we investigated coordination activities as interactional patterns through the socio-cognition lens of collective minding (Cooren 2004), rooted in Weick and Roberts's (1993) collective mind concept. As a result, we present a research framework for investigating how multimodal communication affects collective minding and coordination in virtual teams; we apply this framework to analyze interaction patterns in the considered case; and we present propositions on the impact of multimodal communication on collective minding and coordination performance in virtual teams.

The paper is organized as follows. The next section summarizes the virtual team literature on coordination. Subsequently, we present the framework for investigating the impacts of multimodal communication on virtual team coordination. We then describe our research approach and the investigated case, followed by a detailed account of our analyses. As a result, we discuss empirical insights, theoretical propositions, and research methodology as contributions to the literature on virtual team coordination.

Virtual Team Coordination

Virtual teamwork is characterized by challenges that hinder innovation (Dubé and Robey 2009, Gibson and Gibbs 2006), including coordination challenges (Powell et al. 2004). Coordination is of the utmost importance in any team, but virtual teams often face complex coordination problems due to their distributed nature (Curseu et al. 2008, Gibson and Gibbs 2006). Geographical dispersion requires special attention to coordination due to time-zone differences (Massey et al. 2003, Montoya-Weiss et al. 2001), locally situated knowledge (Sole and Edmondson 2002), and lack of presence awareness (Espinosa et al. 2007). National diversity across teams may imply coordination difficulties related to communication routines (Maznevski and Chudoba 2000), linguistic differences (Kayworth and Leidner 2000), and weak interpersonal relationships (Kraut et al. 1999). Structural dynamics may require special attention to coordination regarding task coupling (Carmel and Agarwal 2001, Ramesh and Dennis 2002, Sakthivel 2005), task awareness (Espinosa et al. 2007), and inter-functional conflict resolution (Robey et al. 2000). Finally, technology mediation may imply coordination difficulties related to limited informal team communication (Herbsleb and Grinter 1999) and organizational identification across teams (Wiesenfeld et al. 1999).

Effective team coordination requires the sharing of knowledge, and current research suggests that it is difficult to exchange and share knowledge across sites (Cramton 2001, Majchrzak et al. 2005, Sole and Edmondson 2002). Virtual team researchers have coined this the mutual knowledge problem (Cramton 2001) or the situated knowledge challenge (Sole and Edmondson 2002). To address these difficulties, it has been suggested to communicate differences in context enabled by information technology (Majchrzak et al. 2005); to focus on how different technologies offer different advantages and disadvantages for enhancing team effectiveness (Kirkman and Mathieu 2005; Hertel et al. 2005); and to explore ways to support virtual team coordination that are close to real working conditions in cross-organizational collaboration (Martins et al. 2004). Along similar lines, Fiore et al. (2003) suggest researchers need to address more fully how the level of media richness associated with collaboration technology influence attitudes, behaviors, and cognitions in virtual teams.

Existing research reveals, in this way, important and persistent challenges related to virtual team coordination (Espinosa et al. 2007, Kanawattanachai and Yoo 2007, Kotlarsky et al. 2008) and it proposes additional investigations into how new and different forms of mediated communication can help address these challenges. Yet, the literature is silent on how virtual teams can coordinate based on multimodal communication. This research has been designed to address this gap by specifically responding to Yoo and Kanawattanachai's (2001) call to investigate the collective mind through micro-level content analysis of mediated interactions in virtual teams.

Research Framework

Coordination Performance

Coordination, a key activity in any organization, is defined as managing dependencies between activities (Malone and Crowston 1994). While virtual team coordination is intrinsically linked to performance (Johansson et al. 1999, Maznevski and Chudoba 2000), it is a challenge because of time zones, cultural divides, divergent mental models, and mediated communication (Galegher and Kraut 1994, Kanawattanachai and Yoo 2007, Kayworth and Leidner 2000, Sarker and Sahay 2004, Warkentin et al. 1997). Successful coordination is characterized by the integration and harmonious adjustment of individual activities towards the accomplishment of a larger goal (Singh 1992) or simply by working together effectively (Malone and Crowston 1991). Because coordination is most clearly noticeable when it is lacking (Malone and Crowston 1994), we investigate coordination performance in virtual teams by analyzing the extent to which it fails.

A communication breakdown causes a disruption in work practices, shifting the actors' attention towards an appropriate recovery strategy (Bjørn and Ngwenyama 2009, Ngwenyama 1998). Communication breakdowns in virtual teams can range from specific instances of failed turn-taking (Garcia and Jacobs 1999, Sarker and Sahay 2003) to conflicts due to the differing perspectives inherent in different organizational roles (Griffin and Hauser 1996, Robey et al. 2000). In the context of virtual teams, communication breakdowns are defined as compromising or challenging coordination between actors on four different levels (Bjørn and Ngwenyama 2009). Lifeworld breakdowns occur when the taken-for-granted constitutive knowledge underpinning the coordination effort is challenged; organization breakdowns occur when existing organizational policies, procedures, technologies, and norms for coordination are challenged; work-process breakdowns occur when the efficacy of teamwork coordination practices and routines is challenged; and technology mediation breakdowns occur when the practical use of communication technology is challenged (Bjørn and Ngwenyama 2009). We analyze these four levels to identify and categorize communication breakdowns during virtual team coordination. Specifically, we assess a virtual team's coordination performance by quantitatively analyzing how many breakdowns occur and by qualitatively analyzing the impact of breakdowns on coordination performance.

Collective Minding

The ability to coordinate and collaborate effectively in organizations has been conceptualized as a socio-cognitive phenomenon through the notion of a collective mind (Weick and Roberts 1993) and similarly through the notions of distributed cognition (Boland et al. 1994, Rogers and Ellis 1994, Wright et al. 2000) and shared cognition (Cannon-Bowers and Salas 2001, Cannon-Bowers et al. 1993, Ensley and Pearce 2001). While developed for the context of high-reliance organizations, a collective mind should also be pursued in other organization types (Cooren 2004, Weick et al. 1999).

The collective mind has been linked to innovation capabilities with information technology (Swanson and Ramiller 2004) and performance in knowledge-intensive teams (Bijlsma-Frankema et al. 2008). The collective mind perspective has, furthermore, provided valuable insights into coordination in software requirements' development (Crowston and Kammerer 1998), board meetings (Cooren 2004), and virtual teams (Yoo and Kanawattanachai 2001). The major claim of collective mind theory is that individuals facilitate group performance by developing shared understandings of a team's tasks and of one another (Crowston and Kammerer 1998). Weick and Roberts (1993) define collective mind as:

> '...' a pattern of heedful interrelations of actions in a social system. Actors in the system construct their actions (**contributions**), understanding that the system consists of connected actions by themselves and others (**representation**), and interrelate their actions within the system (**subordination**).

Heedfulness is an important attribute of the collective mind. Actors behave heedfully to the extent that they engage carefully, critically, consistently, purposefully, attentively, studiously, vigilantly, conscientiously, and pertinaciously (Weick and Roberts 1993). In a collective mind, each act is heedfully interrelated with the acts of other actors. Each actor must subject (subordination), depict (representation), and provide (contributions) for the social system as he or she continuously coordinates activities.

While research into the collective mind of virtual teams is still limited, Yoo and Kanawattanachai (2001) have related the concept to virtual team performance and called for micro-level content analysis of how mediated communication can enable the collective mind. Such analyses have proven valuable in Cooren's (2004) conversation analyses of board meetings. While controversial in some ways (Cooren 2006, McPhee et al. 2006), Cooren's study provides valuable insights into how to investigate the collective mind in organizational communication. The study suggests that evidence of a collective mind has to be found in interactional patterns and not only in the perception of individual actors (Cooren 2004). Cooren (2004) proposes to focus on the process of collective minding, emphasizing that a collective mind can be found at multiple stages of development in an organization. Because actions are the primary units of analysis, and because the interrelation of actions is mediated by information technology, we adopt a language–action perspective to investigate collective minding in virtual teams.

Multimodal Communication

Coordination is the management of dependencies between activities (Malone and Crowston 1994) and understanding the heedful management of these dependencies requires a focus on how coordination is communicated and enacted. The language–action perspective, also known as speech act theory, provides such a focus on how people act through language. Speech act theory was developed by Austin (1962) and Searle (1969) based on the observation that utterances are not necessarily statements whose truth is at stake. Performatives, such as declarations or directives, can be uttered more or less appropriately, but they are not in a simple sense true or false. Similarly, commands, questions, and apologies are not descriptions of a nonlinguistic world (Winograd 1987). Adopting a language–action perspective to investigate how collective minding is mediated through teleconferencing, we apply Searle's (1979) typology of speech acts in cooperative work (Winograd 1987):

- ***Assertive***: Commit the speaker (in varying degrees) to something's being the case – to the truth of an expressed proposition.
- ***Directive***: Attempt (in varying degrees) to get the hearer to do something. These include both questions (which can direct the hearer to make an assertive speech act in response) and commands (which direct the hearer to carry out some linguistic or nonlinguistic act).

- ***Commissive***: Commit the speaker (again in varying degrees) to some future course of action.

- ***Declarative***: Bring about the correspondence between the propositional content of the speech act and reality (e.g. pronouncing a couple married).

- ***Expressive*:** Express a psychological state about a situation (e.g. apologizing and praising).

Articulations of these five speech act types can be modified by the degree of illocutionary force (Holmes 1984, Sbisà 2001), making a statement more or less powerful. Virtual team communication is, however, not limited to articulations through the audio modality. Actors can also act through the visual modality with the use of information technology (Winograd 1987). In the considered case, actors complemented teleconferencing with real-time collaborative modeling based on mindmapping[1] technology. During virtual meetings, actors would collaboratively and in real time manipulate text and other symbols to model a mindmap of their current and future efforts. This technology offered seven different types of manipulations to help actors communicate during virtual meetings: four allowed actors to manipulate a shared mindmap through the creation, deletion, movement, or renaming of nodes; and three allowed actors to add information to nodes through task assignment, node prioritization, and status reporting. As a result, we adopted this combination of articulations in the audio modality and manipulations in the visual modality to investigate virtual team coordination based on the multimodal use of information technology.

Summary

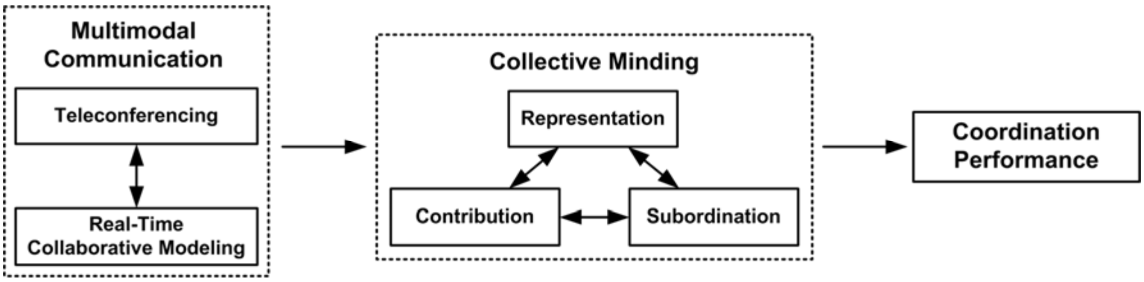

Figure 1 Research framework

Figure 1 summarizes the adopted research framework, drawing on the theories of speech acts (Winograd 1987), communication breakdowns (Bjørn and Ngwenyama 2009), coordination (Malone and Crowston 1994), and collective mind (Weick and Roberts 1993). A virtual team's coordination performance is considered through the lens of collective minding based on multimodal communication. We investigate multimodal communication, based on teleconferencing and real-time collaborative modeling, by analyzing how articulations and manipulations are combined and interact (see Figure 1). We investigate collective minding by identifying how representation, contribution, and subordination are enacted through articulations and manipulations. The virtual team exhibits collective minding to the extent that these acts are heedfully interrelated. Finally, we investigate coordination performance by identifying communication breakdowns, by analyzing how these breakdowns affect coordination, and by assessing virtual team coordination performance in general. Based on the identified gap in current virtual team research and drawing on this framework, we pose the following research question:

[1] A mindmap is a diagram used to represent words, ideas, tasks, or other items linked to a central keyword or idea.

How does multimodal communication based on teleconferencing and real-time collaborative modeling affect collective minding and coordination in virtual teams?

Research Method

A Case Study Approach

We adopted a case study approach for a number of reasons (Benbasat et al. 1987, Yin 2003): our research is guided by a how question; collective minding in virtual team coordination is a contemporary phenomenon that needs further investigation in real-life contexts (Yoo and Kanawattanachai 2001); and, while we know that collective minding affects virtual team performance (Yoo and Kanawattanachai 2001), the relationships between multimodal communication, collective minding, and coordination performance are not well understood. In addition, we had access to a case with unique and interesting characteristics (Yin 2003), where teleconferencing was combined with real-time collaborative modeling during virtual meetings. The team's task was to finalize the development of the mindmapping tool they used to support collaborative modeling. We were thus able to investigate how a virtual team, which was highly dedicated to multimodal communication, managed to coordinate its efforts. Finally, the case provides rare insights into the sensitive and often troublesome coordination between management, marketing, and product development in a virtual team environment (Griffin and Hauser 1996).

The presented case study is explanatory (Yin 2003). We iteratively compared the theoretical research framework in Figure 1 with empirical evidence to explain how the virtual team managed to coordinate successfully with multimodal communication. Similar to Cooren's (2004) conversation analyses of collective minding based on transcripts of a single board meeting, our study provides detailed analyses of multimodal communication during a virtual team's meetings. Access to multiple data sources including tape recordings of all mediated articulations and video recordings of all mediated manipulations provided rich opportunities for combining quantitative and qualitative analyses (Mingers 2001, Sherif et al. 2006).

The Case

The investigated virtual team was a joint venture between a small Danish software company in Copenhagen, *Software.DK*, and a Russian R&D outsourcing provider in St Petersburg, *Software.RU*. *Software.DK* was established in January 2006 by 4 Danish partners who, between them, had 30 years of experience of developing computer simulations and intelligent learning solutions. Previously, they had developed a portfolio of advanced medical micro-simulators based on collaboration with *Software.RU* and other software development outsourcing companies in India. *Software.RU* was established in 1991 and had more than 350 Russian employees. The company had been engaged in more than 300 projects with companies from Denmark, Finland, Germany, Sweden, and the US.

The project was initiated by *Software.DK* in February 2006 with the goal of developing a web-based, collaborative mindmap system to support systems development. The joint venture was established with *Software.RU* in April 2006 and named the *Comapping* project. The two companies had equal ownership, but made different contributions to the project. *Software.RU* initially assigned two developers to the project while *Software.DK* provided management, architectural, and design expertise. With two developers in *Software.DK* initially working full-time on the project along with the two Russian developers, there was a proof of concept ready the following month.

The *Comapping* project shifted its focus and hired three managers to develop a commercial strategy for the new system. The managers were, however, not able to agree on a strategy and were therefore released from the project. It was at this point, in early 2007, that we initiated contact with the *Comapping* project. After a period of technically focused management aimed at finalizing a first, full version of the system, we started to observe systematically all the virtual meetings between the Danish and Russian sites in April 2007. The project was then managed by a team consisting of the CEO of the *Comapping* joint venture, a board member of *Software.DK*, and the Russian systems development manager. Three months later, the project reached a major milestone when a Fortune 500 company invested in the system. As a result, customization to the new partner's requirements became the primary objective and the *Comapping* project staff was increased to eight full-time developers. Our case study ended in August 2007, when this milestone was reached and the project was reorganized.

Coordination in the virtual team relied mainly on teleconferencing via Skype (www.skype.com) combined with real-time collaborative modeling via the mindmapping tool (www.comapping.com). Virtual meetings were held between the *Software.DK* board member (representing management), the joint venture CEO (representing marketing), and the Russian systems development manager (representing product development). The conference language was English and all the virtual meetings took place within normal working hours as the time-zone difference between Copenhagen and St Petersburg is only two hours. The virtual meeting structure was closely reflected in the mindmap tool. Figure 2 shows the mindmap at the start of a meeting and the revised mindmap resulting from the meeting is shown by Figure 3.

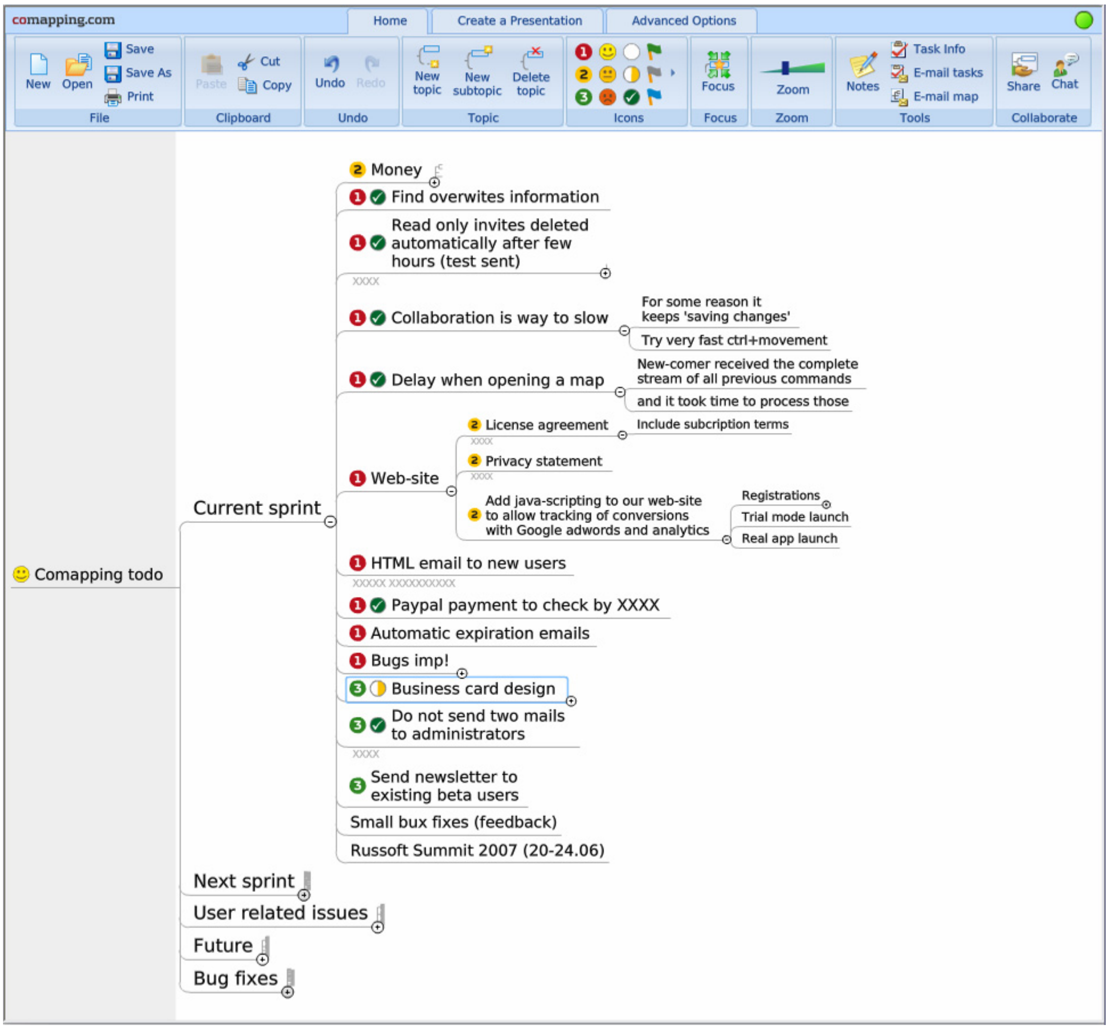

Figure 2 Screenshot of the *Comapping* mindmap before the virtual meeting on May 21, 2007

The root "comapping todo" has two sub-nodes "current sprint" and "next sprint." The sub-nodes represent assignments to individual team members or groups. Assignment priorities are usually indicated by a number and development status with a checkmark or an empty, half-, or three-quarter-full circle. These priority and status indicators are also applied to sub-nodes of the mindmap hierarchy, e.g. "Web-site", see Figure 2. Another frequently used indicator assigns a node to a specific individual using small text boxes below the node (names have been replaced with Xs). When multiple users navigate the mindmap, each individual's cursor is visible to other users as a small box with the name of that individual. Manipulations of the mindmap, e.g. deleting, adding, or changing a node, are instantly made visible to other users.

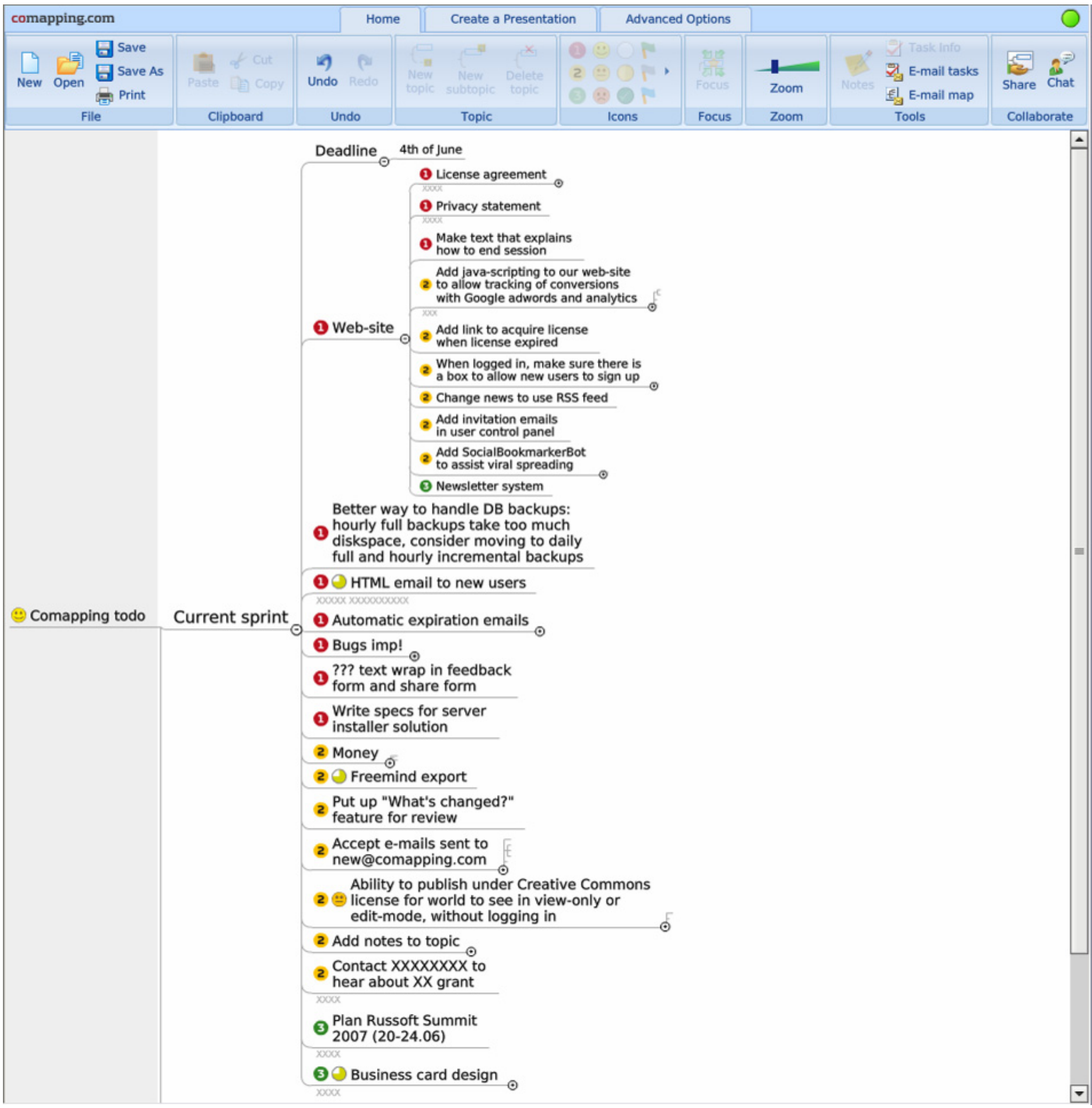

Figure 3 Screenshot of the *Comapping* mindmap after the virtual meeting on May 21, 2007

The virtual meetings usually start with a walkthrough of all the sub-nodes of "current sprint," also known as the sprint backlog, in the agile software development method Scrum (Rising and Janoff 2000, Schwaber and Beedle 2001). The Russian systems development manager would report on the status of each node and nodes are deleted as assignments are completed. Other assignments are elaborated with new sub-tasks or they are given alternative priority or status. New assignments might also be introduced during this initial walkthrough. Virtual meetings would then typically continue with a walkthrough of "next sprint" with sub-nodes representing assignments or ideas postponed for later. This would lead to changing the priorities of some sub-nodes or relocating sub-nodes to "current sprint." By the end of a virtual meeting, the actors would revisit "current sprint" and consider the feasibility of assignments and agree on a deadline for the sprint. The sub-nodes of "current sprint" would also be ordered according to priority, with the highest priorities placed highest.

Data Collection

We collected data about the *Comapping* project from January 2007 through August 2007, including field observations, recordings of the virtual meetings, and team member interviews focused on the wider context. During the virtual meetings, the first author was present offsite as a passive observer, while audio recording conversations and video recording real-time collaborative modeling. A total of seven meetings were observed from April 2007 to July 2007 (see Table 1). The observations were initiated when the marketer was recruited to the project and ended at a major milestone when the Fortune 500 company invested in the system. Even though the mindmapping tool had not been officially released when the data collection started, all its basic functionality was readily available for the team's coordination.

Date (2007)	Duration	Participants
April 24	32 min	Marketer, Manager, Developer
May 3	37 min	Marketer, Manager, Developer
May 16	17 min	Marketer, Manager, Developer
May 21	64 min	Marketer, Manager, Developer
June 4	32 min	Marketer, Manager, Developer
June 26	50 min	Marketer, Manager, Developer
July 2	15 min	Marketer, Manager, Developer

Table 1 Virtual meetings in the *Comapping* project

Before we started observing virtual meetings, we conducted semi-structured interviews to help us understand the organization and work-group contexts (Majchrzak et al. 2000). These interviews were initiated with a face-to-face meeting with a key member of the *Comapping* project followed by interviews with other project members via Skype. Towards the end of our observations, we conducted a new series of interviews with key project members and the CEO of *Software.DK*. In total, we conducted eleven interviews (see Table 2).

Site and Nationality	Role	Background	Date (2007)
Denmark	**Manager** Board Member of the *Comapping* project and Director of Technology & Innovation at *Software.DK*	Co-founder of *Software.DK* in 2006 and earlier co-owner of a software company in which he was Director of Research & Technology with responsibility for managing global projects and outsourcing relationships based on agile approaches. Education: Computer Science	January 19 March 6 April 24 June 27
Russia	**Developer** Director of R&D for the *Comapping* project	Manager of an Offshore Development Center within the Russian outsourcing provider for several years. Education: Computer Science	March 22 July 13
Russia	Software developer in the *Comapping* project	Three years of experience as software developer at the Russian outsourcing provider. Education: Physics and Mathematics	April 5 July 17
Denmark	**Marketer** CEO for the joint venture	Four years of experience as strategy consultant at major consultancy firms. Specialized within front-end optimization within marketing and sales ranging from segmentation and pricing to best practice and process optimization. Education: International Management	May 8 June 28
Denmark	Chairman of the *Comapping* joint venture, and CEO of *Software.DK*	Co-founder of *Software.DK* in 2006 and earlier co-owner of a software company in which he was Global Director of Learning Products. Education: Medicine and Medical Education	August 8

Table 2 Semi-structured interviews

Data Analyses

We analyzed the data using Atlas.ti V5.5 (Muhr 2008), allowing us to code virtual meeting recordings directly. The transcription of recordings was not a feasible approach because of the high interrelation between audio and visual modalities in the data. We coded the seven virtual meetings and used the data from the eleven semi-structured interviews to help us understand the context, antecedent conditions, and outcomes.

Table 3 summarizes the adopted coding scheme. We used three constructs – articulation acts, manipulation acts, and communication breakdowns – to find evidence of the concepts in the research framework (see Figure 1): multimodal communication via teleconferencing and real-time collaborative modeling; collective minding defined through representation, contribution, and subordination; and coordination performance. While all the constructs have significance for all the concepts in the research framework, each construct is primarily related to two of the concepts (see Table 3). Articulations were identified in the teleconferencing data and we considered them primary indicators of the actors' contributions to collective minding. Manipulations were identified in the real-time collaborative modeling data and we considered them primary indicators of the actors' representations in collective minding. Finally, communication breakdowns were identified based on both the articulation and manipulation data. These breakdowns convey a lack of ability to interrelate actions during virtual meetings and we considered them primary indicators of coordination performance and actors' subordination to collective minding.

Construct	Subcategory	Description	Primary Signification
Articulation Act	Assertive	Commit the speaker (in varying degrees) to something's being the case – to the truth of the expressed proposition.	**Teleconferencing**
	Directive	Attempt (in varying degrees) to get the hearer to do something. These include both questions (which can direct the hearer to make an assertive speech act in response) and commands (which direct the hearer to carry out some linguistic or nonlinguistic act).	
	Commissive	Commit the speaker (again in varying degrees) to some future course of action.	**Contribution**
	Declarative	Bring about the correspondence between the propositional content of the speech act and reality (e.g. pronouncing a couple married).	
	Expressive	Express a psychological state about a situation (e.g. apologizing and praising).	
Manipulation Act	Create node	Making a node in the mind map.	**Real-Time Collaborative Modeling**
	Delete node	Removing a node from the mind map.	
	Move node	Relocating a node with or without sub-nodes in the mind map.	
	Rename node	Changing the text in a node in the mind map.	
	Prioritize node	Creating or changing a numeral attached to a node in the mind map.	**Representation**
	Assign task	Creating or changing a textual attachment with one or more names to a node in the mind map.	
	Report status	Creating or changing a symbol attached to a node in the mind map that is either a checkmark or an empty, half- or three-quarter-filled circle.	
Communication Breakdown	Lifeworld	Challenging the taken-for-granted constitutive knowledge underpinning the coordination effort	**Coordination Performance**
	Organization	Challenging the established organizational policies, procedures, technologies, and norms for coordination	
	Work process	Challenging the team's coordination practices and routines	**Subordination**
	Technology mediation	Challenging the practical use of technology to mediate coordination	

Table 3 Coding scheme

The first author initially coded articulations and manipulations in the longest virtual meeting. The second author then critiqued the coding, leading to minor improvements of the coding scheme. The scheme was then shared with a research assistant, who, along with the first author, recoded the longest virtual meeting. In the case of disagreements, coding options were discussed between the 2 coders until agreement was reached. There were no disputes in 90% of the instances. The second author then reviewed the coding of the longest virtual meeting by evaluating 5 randomly chosen instances of each articulation subcategory. Only 1 articulation instance under the Declaration subcategory led to disagreement, indicating 96% (1/25) agreement. All the virtual meetings were then coded according to the coding scheme by the first author and the research assistant.

The coding of communication breakdowns in virtual meetings was initiated by introducing the coding scheme in Table 3 to the research assistant. The first author and assistant then coded 3 meetings. In cases of disagreement, the options were discussed until an agreement was reached. In this first step, 45% of the coded instances initiated no dispute between the 2 coders. The second author reviewed the coding and that led to a more inclusive interpretation of communication breakdowns. As a second step, the first author and the research assistant coded 2 additional meetings where 65% of the instances caused no dispute between the coders. Once again, the second author reviewed the coding, which led to 2 recategorizations and 1 deletion out of 22 identified communication breakdowns (86% agreement rate between the 2 coders and the second author). The review also led to further clarification of the coding scheme. In a final step, the first author and assistant coded all the virtual meetings from scratch following Table 3. In this process, 90% of the coded instances initiated no dispute between the 2 coders. In cases of disagreement, the options were discussed until agreement was reached. Finally, the actor(s) who primarily triggered and alleviated each breakdown was identified.

We systematically triangulated the analyses of audio and video recordings of the seven meetings with the interview data presented in Table 2. The interviews were recorded and revisited multiple times throughout the analysis in order to establish antecedent conditions and outcomes, to address ambiguities in our analyses, and to compare our findings with the actors' perceptions. In this way, we related our analyses of the collaborative technology in the *Comapping* project to the organizational and work-group contexts (Majchrzak et al. 2000).

Results

Quantitative Analyses

Initially, we examined the data quantitatively based on the framework in Figure 1 and the constructs in Table 3. Table 4 and Table 5 summarize the articulations and manipulations during virtual meetings in the *Comapping* project and Table 6 summarizes the communication breakdowns and how actors triggered and alleviated them.

	Manager	Developer	Marketer	Total
Assertive	72	114	39	225
Commissive	27	54	58	139
Declarative	7	2	5	14
Directive	72	21	74	167
Expressive	39	29	26	94
Total	217	220	202	639

Table 4 Distribution of articulation acts in the audio modality

Overall, we identified 639 articulations corresponding to 2.59 (639/247) per minute. The articulations were equally distributed across the actors while the distribution of each articulation type differed significantly (see Table 4). Assertive acts were most frequently performed by the developer and least by the marketer; commissive acts were least frequently carried out by the manager; declarative acts were relatively rare; directive acts were less frequently performed by the developer; and expressive acts were almost evenly distributed.

	Manager	Developer	Marketer	Total
Assign task	7	1	3	11
Create node	39	5	11	55
Delete node	20	10	8	38
Move node	30	1	1	32
Prioritize node	22	4	3	29
Rename node	12	0	3	15
Report status	9	8	6	23
Total	139	29	35	203

Table 5 Distribution of manipulation acts in the visual modality

Overall, we identified 203 manipulations corresponding to 0.82 (203/247) per minute. The manipulations were unequally distributed across the actors (see Table 5). The manager was responsible for 69% (139/203), the marketer for 17% (35/203), and the developer for 14% (29/203) of the manipulations of the mindmap. The most evenly distributed manipulation act was report status while the most unevenly distributed was move node.

		Lifeworld	Organization	Work Process	Technology Mediation	Total
All Actors		11	7	33	10	61
Actors triggering communication breakdowns	**Manager**	3	0	6	2	11
	Developer	3	1	9	1	14
	Marketer	4	5	17	2	28
	No Actor	2	1	1	7	11
Actors alleviating communication breakdown	**Manager**	6	5	18	7	36
	Developer	6	4	14	5	29
	Marketer	3	1	8	2	14

Table 6 Distribution of communication breakdowns

We identified a total of 61 breakdowns, corresponding to 0.25 (61/247) per minute. The most frequently occurring breakdown type was related to work process and accounted for 54% (33/61) (see Table 6). The remaining breakdowns were equally distributed between lifeworld, organization, and technology mediation issues. The marketer contributed most frequently to triggering organization and work-process breakdowns; the three actors contributed more equally to lifeworld breakdowns; and technology mediation breakdowns were most frequently, 70% (7/10), triggered by technology issues rather than a specific actor. The manager and developer contributed most frequently, 82% (65/79), to alleviating communication breakdowns.

Qualitative Analyses

As a next step, we investigated the coordination performance in the *Comapping* project qualitatively, focusing on how coordination was challenged through communication breakdowns. For each communication breakdown type, we reviewed relevant aspects of the quantitative analyses; we conducted detailed analyses of exemplar breakdowns; and we analyzed related participant perceptions based on interviews with virtual team members.

Technology Mediation

Technology mediation breakdowns occurred when the practical use of teleconferencing and real-time collaborative modeling was challenged. This type of breakdown was rare and technology issues rather than actors triggered 70% (7/10) of them (see Table 6). Table 7 provides an overview of all the technology mediation breakdowns. The technical difficulties were poor sound quality, network connection failure, error in the mindmap system, and erroneous participation representation in the mindmap; the breakdowns caused by actors involved a lack of attention to technology features and problems related to turn-taking.

Incident	Occurrences
Poor sound quality	3
Network connection failure	2
Error in the mindmap system: slow update of manipulation in the mindmap	1
Erroneous participation representation in the mindmap	1
Attention to how to operate the real-time collaborative modeling tool	1
Attention to e-mail errors	1
Failed turn-taking among participants	1

Table 7 Technology mediation breakdowns

While turn-taking is considered a challenging and significant issue in virtual team communication (Garcia and Jacobs 1999, Sarker and Sahay 2003), there was only one turn-taking breakdown during the *Comapping* project meetings. This incident involved the manager and marketer (see Quote 1), and it included one manipulation, assign task (Line 5), and three articulations, two directive (Lines 2 and 5) and one commissive (Line 3).

1 **Concurrently:**

__Manager__ (Denmark): Then we have ...

__Marketer__ (Denmark): I also ...

[Eight seconds pause]

2 *__Manager__ (Denmark): ... the website*

3 *__Marketer__ (Denmark): The website, I think, the help button, I will do that.* [Marketer moves his marker to the node "Web site" and then to its sub-node "Help"]

4 *__Manager__ (Denmark): Ahha ...* (Acknowledging)

5 *__Marketer__ (Denmark): That should be moved up.* [Marketer assigns the task "Help" to himself]

Quote 1 Virtual meeting, April 24, 2007

In Line 1, the manager and marketer articulate simultaneously, causing a period of silence. In the recovery of the breakdown in Line 2, the manager makes a directive act, expressed through the shared reference point, "Web site," in the mindmap. By referring to a specific node, the manager brings immediate attention to what he intends to communicate and reduces the likelihood of misunderstandings. The marketer then repeats the manager's statement in Line 3 and places his marker on the "Web site" node. The marketer continues with a commissive act in Line 3 supplemented by the manipulation act in Line 5. Quote 1 illustrates how the manager and marketer quickly recovered from the turn-taking breakdown by combining articulations and manipulations mediated by the two technologies. This pattern of combining articulation and manipulation occurred quite frequently during the *Comapping* project meetings. In total, 42% (86/203) of manipulations were combined with articulations.

The heedful communication in Quote 1 provides evidence of collective minding. While failed turn-taking can indicate a lack of heed during mediated communication in virtual teams, the breakdown in the *Comapping* project had a limited adverse effect on the communication between the manager and marketer. Moreover, the two actors exhibited heedfulness, when exploiting both communication modalities to recover immediately from the breakdown. The actors subordinated their articulations and manipulations to the shared representation readily available in the mindmap. Interestingly, a collective mind can degenerate if it is unchallenged for a longer period of time (Weick and Roberts 1993). So, besides providing evidence of how actors recovered from a communication breakdown, Quote 1 shows how the collective mind was revitalized through actors' heedful responses to breakdowns. In total, we only identified 1 turn-taking breakdown in 203 manipulations and 639 articulations during 7 virtual meetings and 4 hours of activity. This indicates high coordination performance in the *Comapping* project as failed turn-taking is a common challenge in mediated communication (Garcia and Jacobs 1999, Sarker and Sahay 2003).

All three actors in the virtual meetings had previous experience of working in virtual teams. In an interview, the marketer argued for the adopted approach to collaborative mindmapping by critiquing communication technology choices such as teleconferencing and e-mail in his past virtual team experiences. Hence, the actors' past experiences of working in virtual teams influenced their perceptions and capabilities during the considered multimodal communication in the *Comapping* project.

Work Process

Work-process breakdowns occurred when the efficacy of the team's coordination practices and routines were challenged. This type of breakdown was the most frequent, constituting 54% (33/61) of all the breakdowns during the meetings (see Table 6). Table 8 provides an overview of the individual incidents. The most frequent breakdowns were "participants request repeat of articulation" and "uncertainty concerning how to use the real-time collaborative modeling tool." These breakdowns account for 38% (23/61) of all the communication breakdowns during the meetings.

Incident	Occurrences
Participants request repeat of articulation	12
Uncertainty concerning how the team uses the real-time collaborative modeling tool	11
Participants reveal misrepresentation of information articulated by another participant	6
Talking on the phone during the virtual meeting	2
Failed coordination of mindmap manipulations	1
Misrepresentation of information in the mindmap	1

Table 8 Work-process breakdowns

The most frequent incident is "participants request repeat of articulation," constituting 20% (12/61) of all breakdowns. This is a very common conversational breakdown and it is interesting to investigate how the actors responded in the *Comapping* project. Quote 2 presents one typical conversational breakdown involving the manager and the developer. The quote includes two manipulations, the create node (line 3) and prioritize node (Line 6), and four articulations, two directives (Lines 1 and 2), one assertive (Line 4), and one declarative (Line 6).

1 ***Manager*** *(Denmark): Another thing, when you log in, it should also have the box for signing up a new user.*

2 ***Developer*** *(Russia): When you log in ... say it again.*

3 ***Manager*** *(Denmark): I am just writing it up under website ... When logged in, make sure there is a box to allow new users to sign up.* [Manager creates node as a sub-node under "Web site"]

4 ***Manager*** *(Denmark): So, if I am using my computer and would like to sign someone else up there is no way I can do that right now without logging out.*

5 ***Developer*** *(Russia): Ahhh ... Okay ...*

6 ***Manager*** *(Denmark): So, that is a huge bug.* [Manager prioritizes node on level 2]

Quote 2 Virtual meeting, May 21, 2007

The manager articulates a directive in Line 1. However, the developer appears inattentive and articulates in Line 2 a directive, requesting the manager to repeat his articulation. In response, the manager repeats his initial directive in Line 3 while also creating a node. This is visible in Figure 3 where, in comparison with Figure 2, all the changes of this particular virtual meeting are represented. Then, the manager emphasizes the directive with an assertive articulation in Line 4. The developer acknowledges in Line 5 and the manager declares that it is a huge bug in Line 6. Quote 2 illustrates how the manager heedfully alleviates the breakdown by combining articulations and manipulations mediated by the two technologies. The pattern of combining articulations and manipulations is similar to the one in Quote 1. However, Line 6 in Quote 2 illustrates an interesting inconsistency between the manager's articulations and manipulations. While boosting the illocutionary force of the articulation (Holmes 1984, Sbisà 2001) by stating it is "a *huge* bug," the manager sets the node priority to level 2. So, in this case, the declarative use of articulations was modified by a manipulation, which provides one possible reason why only 2% (14/639) of the articulations were declarative.

The developer's request for repeating an articulation indicates a possible lack of heed. However, the breakdown enhances the manager's heedfulness as he immediately exploits both communication modalities to alleviate the breakdown. Interestingly, this shows how an actor in the *Comapping* project exhibited heedfulness as he perceived another member of the collective mind as heedless. This observation is in contrast to the general claim that heedless interrelating can induce further heedlessness in the organization (Weick and Roberts 1993). One possible explanation is that the incident of heedless interrelating by the developer had little significance for the manager because the developer immediately explicated his lack of heed (Line 2). The manager's representation of the social system was as a result not affected, in contrast to situations in which heedlessness by an actor might have ripple effects on the level of heed exhibited by other actors because breakdowns are created in each individual's representation of the social system.

The *Comapping* project was not only supported by representations in the mindmap. In an interview, the manager argued that e-mail notifications of code contributions sent to all the project members served as a simple but important shared representation of the project. These e-mail notifications were systematically reviewed by both the developer and the manager, enabling a detailed representation of the progress and quality of the project members' contributions. The manager also argues that, while everyone can make mistakes, their work process entailed the early discovery of most errors, avoiding later cost escalation. A similar frequent review process was conducted when code was transferred from the test server to the production server. The manager also regularly coordinated directly with the developers in Russia without involving the managing developer, further entailing a detailed project representation. In his account of these activities, the manager emphasized the importance of continued high attention to quality in even the smallest detail, which when enacted is mirrored by the other actors in the *Comapping* project.

Organization

Organization breakdowns occurred when existing organizational policies, procedures, technologies, and norms for coordination in the *Comapping* project were challenged. This type of breakdown was the least frequent, constituting only 11% (7/61) of the incidents (see Table 6). The marketer, who was the most recent member of the *Comapping* project, contributed to triggering 71% (5/7) of these breakdowns. Table 9 provides an overview of all the organization breakdowns.

Incident	Occurrences
Unclear procedures for business strategies	1
Unclear responsibilities for documentation of agreements	1
Norms of efficiency in the project are challenged	1
Inability to recall previous undocumented agreements	1
Unclear responsibilities for paying fees to external party	1
Participants focus on what should be discussed in the technical focused virtual meetings	1
Undecided procedures for server-upgrading	1

Table 9 Organization breakdowns

One organizational breakdown triggered by the marketer was "unclear procedures for business strategies." This particular incident unfolded as a debate over 6 minutes, making it the most time-consuming breakdown during the observed meetings. The incident illustrates the difficulties in coordinating between development and marketing (Griffin and Hauser 1996). In the following, we analyze the initial part of this incident involving the marketer, manager, and developer (see Quote 3). The quote includes two articulations by the manager, one directive (Line 8), and one expressive (Line 10).

1 ***Marketer*** *(Denmark): "*Developer,*" we talked about the desktop application.*

2 ***Developer*** *(Russia): Yeah.*

3 ***Marketer*** *(Denmark): And also "*Manager,*" we were thinking that the desktop application should be launched no later than two months.*

4 ***Manager*** *(Denmark): No later or not earlier?*

5 ***Marketer*** *(Denmark): Well, I have told people, well only a few. I was thinking we should launch it between one or two months from today's date.*

6 ***Manager*** *(Denmark): Well, hehe ...* (Short laugh).

7 ***Marketer*** *(Denmark): Is that possible? I'm just thinking how should we integrate this into the current sprint?*

8 ***Manager*** *(Denmark): I think that you need to stop telling people when things will come, before we have decided.*

9 ***Developer*** *(Russia): Hehe ...* (Laughing)

10 ***Manager*** *(Denmark): We don't know the impact of the desktop application. We don't know how long it will take yet.*

Quote 3 Virtual meeting, May 21, 2007

In this part of the incident, the marketer refers in Line 1 to a previous discussion with the developer regarding a desktop application version of the *Comapping* product. He also refers to a previous discussion of a deadline in Line 3 and points out that he had shared this information with other people in Line 5. The marketer thereby refers to a commissive articulation he made outside the virtual meeting context. Almost concurrently with the manager's short laugh in Line 6, the marketer in Line 7 asks whether this goal is possible and how they can integrate it into the current sprint. In a response, the manager articulates a directive in Line 8, stating that the marketer should stop revealing such information to people without coordination with the team. The developer briefly laughs, suggesting disagreement with the marketer; this disagreement becomes more pronounced later during the incident. The manager then reasons his directive with the expressive statement in Line 10. The absence of articulation acts throughout the marketer's communication expresses a high level of uncertainty on his behalf. As a consequence, he agrees by the end of the incident that the team should not prioritize the desktop application. (The marketer appears to assert in Line 5 and direct in Line 7; but we did not consider these as articulation acts because of their rather weak illocutionary force (Holmes 1984, Sbisà 2001)).

The marketer's commissive articulation to external parties preceding Quote 3 never became an interrelated contribution as part of a shared representation of the *Comapping* project. While the marketer's referral to a discussion with the developer in Line 1 suggests interrelatedness between the two, his attempt to include the desktop application in the current sprint in Line 7 was unsuccessful. This lack of interrelatedness on the desktop application issue was explicitly pointed out by the manager in Line 8, and, later in the incident, the developer also challenged the marketer's representation in Line 3. Hence, the marketer's contribution to the *Comapping* project related to the desktop application was never subordinated, despite his attempt to do so.

The part of the incident following Quote 3 focused on how to prioritize *Comapping* features from a business point of view. In an interview, the manager pointed out that task prioritization had been a significant challenge between him and the developer during the start of the *Comapping* project. Back then, they maintained a document with a list of prioritized tasks. However, they ended up with a large number of tasks with first priority, and it was difficult for the developer to prioritize between them. Subsequently, the project adopted time boxing in the form of sprints (Jalote et al. 2004, Rising and Janoff 2000) and started to use the mindmapping tool. Project tasks were in this way structured and prioritized in a straightforward fashion and explicitly represented in the mindmap, allowing actors to contribute and subordinate more easily.

While the chairman of the *Comapping* joint venture described the project as having two timelines, one for product development and one for marketing, the mindmap was predominantly related to product development with little consideration of marketing. Against this background and given that the marketer had joined the project recently, it is not surprising to find the interrelation difficulties in Quote 3. However, the Russian developer not participating in the meetings stated that the marketer had a revitalizing effect on the *Comapping* project. The marketer's pressure to move forward, even beyond team capabilities, gave this developer a more positive perception of the project.

Lifeworld

Lifeworld breakdowns occurred when taken-for-granted constitutive knowledge underpinning the coordination of the *Comapping* project was challenged. This type of breakdown constituted 18% (11/61) of the incidents and was the only type evenly triggered by the marketer, manager, and developer (see Table 6). Lifeworld breakdowns were predominantly related to a need for conveying taken-for-granted professional or cultural knowledge. Table 10 provides an overview of all the lifeworld breakdowns.

Incidence	**Occurrences**
A need for conveying fundamental professional knowledge	4
A need for conveying fundamental cultural knowledge	4
Ambiguous language use	1
Uncertainty regarding name articulation	1
Unawareness of the physical location of a new participant	1

Table 10 Lifeworld breakdowns

One breakdown related to "a need for conveying fundamental cultural knowledge" involved an effort to explain a technical requirement based on a particular use case. However, differences in contextual knowledge about the use case triggered a dispute between the developer and the manager. Quote 4 illustrates part of this incident; the quote includes one assertive articulation by the developer in Line 8.

1 ***Manager*** *(Denmark): So it is just if you go to a public library and you log in without having remember me clicked you don't want the next person to be able to access your account.*

2 ***Developer*** *(Russia): Yeah of course but in a public library you usually have to log in to the machine.*

3 ***Manager*** *(Denmark): Well, no not here, it could be anywhere, it could be an Internet cafe, whatever.*

4 ***Developer*** *(Russia): I see ... But I think all these guys usually have some kind of user session.*

5 ***Manager*** *(Denmark): No, it doesn't matter if I go to some computer no matter where it is and I have just logged in without having a special system and I don't click remember me, the next person should not be able to access my account.*

6 ***Developer*** *(Russia): Then just close the browser.*

7 ***Manager*** *(Denmark): Yes, I understand that then if that is what you need to do we should put a notice about that.*

8 ***Developer*** *(Russia): Probably, yes, but we can do nothing about it you understand ...*

Quote 4 Virtual meeting, May 21, 2007

The part of the incident preceding Quote 4 concerns a sub-node of the next sprint called "Log out on close if remember me has not been checked." The developer follows the established routine by accounting for the status and challenges related to the task represented by the node. However, the manager disputes the developer's account of the task's key challenge. In an attempt to communicate his perspective more clearly, the manager refers to an exemplary use situation at a public library in Line 1. However, the developer challenges the circumstances of the exemplary use situation in Line

2. In response, the manager states that the challenge of circumstances isn't valid in his lifeworld and instead refers to another exemplary use situation in Line 3. Again challenged by the developer in Line 4, the manager further elaborates his concern in Line 5. The developer suggests a solution in Line 6, which the manager states should be communicated in the product in Line 7. The developer then makes an assertive articulation in Line 8 elaborating the reasons why no technical solutions are available to address the concerns raised by the manager.

In Quote 4, the actors express different perceptions of the exemplary use situation grounded in their Danish and Russian lifeworlds. While the exemplary use situation seeks to ease difficulties in communicating requirements, it causes a breakdown because of differences in the team members' lifeworlds. However, the manager quickly alleviates the breakdown by shifting to a different reference point. He thereby limits the need for subordination to social systems grounded in the actors' respective Russian and Danish contexts. Uncovering of lifeworld differences was, however, also important to the *Comapping* product as the project was not only targeting a specific region and therefore needed to take regional variations into consideration.

In our analyses of the *Comapping* project, we considered all articulations that were not task-related as coordination breakdowns. While such articulations can be attempts to share knowledge required as a prerequisite for task execution, we saw them as distractions from task-focused coordination. Two such incidents related to conveying fundamental cultural knowledge were triggered by the developer's account of Russian national holidays, when deciding on a deadline for a sprint. While sharing this contextual knowledge did not cause any dispute between team members, it did take up additional project time. Effective coordination with limited sharing of contextual knowledge may require significant trust between actors (Jarvenpaa and Leidner 1999, Jarvenpaa et al. 2004, Kanawattanachai and Yoo 2002); and Weick and Roberts (1993) state that reliable performance may require a well-developed collective mind in the form of a complex, attentive system tied together by trust. According to the chairman of the *Comapping* joint venture and the manager, a high level of trust had been established through past collaboration between the two companies before the joint venture.

Coordination based on limited sharing of professional knowledge across sites also requires significant trust. Only 36% (4/11) of the lifeworld breakdowns in the *Comapping* project pertained to a need for conveying fundamental professional knowledge. In one incident, the marketer stated that he didn't understand the issue being discussed and didn't expect he had to, with which both the manager and the developer agreed. This is in line with Cannon-Bowers and Salas's (2001) suggestion that high performance teams can have systems and tasks so complex that it is impossible for any single team member to hold all the required knowledge. In such cases, professional knowledge is specialized and distributed, requiring coordination between several team members.

Discussion

Drawing on the theories of speech acts (Winograd 1987), communication breakdowns (Bjørn and Ngwenyama 2009), coordination (Malone and Crowston 1994), and collective mind (Weick and Roberts 1993), we developed the research framework in Figure 1 to investigate the research question: *How does multimodal communication based on teleconferencing and real-time collaborative modeling affect collective minding and coordination in virtual teams?* Based on our analyses of the *Comapping* project, we first discuss key empirical evidence that teleconferencing and real-time collaborative modeling facilitated collective minding and contributed to successful outcomes. Subsequently, we draw on the empirical evidence to offer theoretical propositions that explain how multimodal communication can facilitate collective minding and positively impact on coordination performance in virtual teams. Finally, we argue that the presented research framework can support further investigations into the impacts of multimodal communication on virtual team coordination based on the constructs of articulation acts, manipulation acts, and communication breakdowns.

Empirical Findings: The Comapping Project

Discussing the key empirical findings from the *Comapping* project, we follow the logic of the research model in Figure 1, starting with multimodal communication. During the virtual meetings, the actors negotiated the specification of tasks, their priority, and the time box in which they should be addressed. These interrelated activities were mediated by teleconferencing and real-time collaborative modeling. Communication with one technology relied heavily on the mirroring of content by the other technology (cf. Quote 1 and 2), thus enhancing turn-taking capabilities (Garcia and Jacobs 1999, Sarker and Sahay 2003) and modifying articulation force (Holmes 1984, Sbisà 2001). Overall, articulations through teleconferencing were equally distributed across the three actors while the distribution of each type of articulation differed significantly (cf. Table 4). Manipulations of the shared model were unequally distributed across the three actors, with the manager responsible for a significant majority of manipulations (cf. Table 5).

Considering how multimodal communication practices affected collective minding, we draw on the three defining concepts: contribution, representation, and subordination (cf. Figure 1). Teleconferencing offered easy and frequent procurement of contributions during the seven virtual meetings. However, the communication of a contribution was typically reinforced by a subsequent representation in the mindmap (cf. Quote 2). As the manager was by far the most dominant manipulator of the mindmap, he had a powerful role in continuously shaping the shared model of the *Comapping* project (cf. Table 5). Contributions were at times initiated as manipulations of the model, however not without subsequent articulations through teleconferencing.

The actors' representations of the system of interdependent actions in the *Comapping* project were continuously shaped and maintained through a combination of articulations through teleconferencing and manipulations of the shared model in the mindmap. Each actor articulated personal representations of the project, significantly influenced by their role as marketer, manager, and developer (cf. Quote 3) (Griffin and Hauser 1996). As key representations were shared through manipulations in the modeling tool, it became transparent how individual actors' representations were related and whether what was articulated was consistent with what had previously been agreed upon. In this way, collaborative modeling helped the actors immediately identify differences and inconsistencies in representations and supported subsequent resolution through teleconferencing.

Subordination to the evolving system of interdependent actions in the *Comapping* project was evidenced by relatively rare communication breakdowns that in all cases were effectively alleviated (cf. Table 6). The actors furthermore reduced the need for subordination to social systems grounded in their respective Russian and Danish contexts (cf. Quote 4). The modeling tool provided the actors with a comprehensive overview of the system, through which they continuously interrelated their actions. The tool offered a simplified and shared model of the virtual team's goals, tasks, and priorities, a model that was updated in real time and was ready-at-hand during meetings (cf. Quote 1). This up-to-date and comprehensive model of core dependencies in the *Comapping* project helped the three actors keep the project on track as they continuously subordinated to essential project issues and commitments.

The analyses of articulations, manipulations, and breakdowns in the *Comapping* project provide strong evidence of heedful interrelation of actions between the marketer, manager, and developer. Most importantly, there was a low frequency of breakdowns; these breakdowns never severely disrupted the coordination efforts as the actors immediately managed to address them and return to the coordination issues at hand; and, in situations where breakdowns showed evidence of heedlessness by some actors (cf. Quote 2), other actors reinforced heedful interrelation by combining the two communication modalities to alleviate the breakdown immediately. Overall, the analyses demonstrate how the combined use of audio and visual modalities enabled collective minding between the three actors in the *Comapping* project.

Finally (cf. Figure 1), there was high coordination performance in the *Comapping* project in the sense that the marketer, manager, and developer under the considered stage of the project integrated and harmoniously adjusted individual work efforts to accomplish project goals (Singh 1992). Before the start of this stage, the project had focused on technically finalizing the system based on an initial prototype; now, the focus was on preparing the system for the market and on attracting commercial interest. Also, at this point, the marketer joined the management team as a new member to emphasize the commercial goals. High coordination performance in the project was evidenced by the low frequency of breakdowns (Malone and Crowston 1994) (cf. Table 6), and by the effective use of audio and visual modalities to alleviate breakdowns immediately. Moreover, at the end of the considered stage, the team successfully negotiated a strategic product development and marketing alliance with a Fortune 500 company.

Theoretical Reflections: Collective Minding in Virtual Teams

The collective mind concept was developed in the complex, but homogenous, context of an aircraft carrier's flight deck (Weick and Roberts 1993). This is quite different from a virtual team context characterized by inhomogeneous physical work environments and situated lifeworlds (Gibson and Gibbs 2006). It is therefore important to compare and contrast the two contexts. First, the physical environment on the flight deck constitutes the space in which the task is to be coordinated; this physical environment offers a shared representation of the coordination space that can be readily shared amongst the involved actors. Similarly, the virtual environment of a distributed team constitutes the space in which the task is to be coordinated; however, the sharing of this space needs to be mediated through the use of information technology. Second, the onshore lifeworlds of the aircraft carrier personnel are of secondary importance in relation to the task faced on the flight deck; this is continuously reinforced as the actors on the aircraft carrier emerge into the same physical work environment as long as the carrier is offshore. Similarly, the local lifeworlds of the *Comapping* project members are of secondary importance in relation to the task faced by the virtual team; however, the actors emerge into different lifeworlds and need to be continuously reminded of their shared commitments and activities in the virtual space. Simply put, mediated communication therefore needs to play a central role to enable collective minding in a distributed team environment.

To help us understand and further investigate how virtual team members can heedfully interrelate through mediated communication, we present four theoretical propositions based on the findings from the *Comapping* project. The propositions explain how multimodal communication can facilitate collective minding and impact on performance in virtual teams through attention to technology, work processes, organization, and lifeworlds (Bjørn and Ngwenyama 2009). We discuss each proposition in relation to the collective minding and virtual team coordination literature.

Proposition 1 (Technology mediation). *Multimodal communication based on teleconferencing and real-time collaborative modeling can facilitate collective minding in virtual teams and positively impact on coordination performance.*

The analyses of the *Comapping* project demonstrate how the combined use of teleconferencing and real-time collaborative modeling enabled collective minding between the marketer, manager, and developer. The detailed analyses of articulations, manipulations, and breakdowns during the seven virtual meetings provide evidence of the heedful interrelation of actions between the three actors. Most importantly, there was a low frequency of breakdowns and breakdowns never severely disrupted the actors' attention to the coordination issues at hand. In addition, the project was successful in meeting its stated objectives. These findings motivate Proposition 1 on technology mediation in virtual team coordination.

Weick and Roberts (1993) provide examples of underdeveloped collective minding such as groupthink (Galanter 1989), the *Challenger* disaster (Starbuck and Milliken 1988), and ethnocentric research groups (Weick 1983). These examples share subordination to a social system that is envisaged carelessly. The combination of teleconferencing and real-time collaborative modeling in the *Comapping* project helped the actors heedfully negotiate and maintain a shared representation of the project. The ease of contributing through teleconferencing and a shared, real-time model provided the *Comapping* project with flexibility. Such flexibility facilitates collective minding, as organic systems typically have more fully developed minds than mechanistic systems because of their capacity to reconfigure themselves into dynamically shifting structures (Weick and Roberts 1993). In addition, smart systems do the right thing regardless of their current structure and of whether the environment is stable or turbulent (Weick and Roberts 1993). We should therefore expect actors' approaches to coordination to affect mediated team coordination, in particular when enabled by combinations of different information technologies. Other case studies have suggested that leaner communication media such as e-mail can facilitate collective minding (Im et al. 2005). However, such leaner media will likely require considerable efforts to construct a heedful project representation; they will likely lead to more frequent and severe communication breakdowns; and as a result the project may be more likely to be envisaged carelessly.

> ***Proposition 2 (Work process).*** *The combination of teleconferencing and real-time collaborative modeling can help actors heedfully sense and respond to work-process breakdowns in virtual team coordination.*

There were on average 0.25 communication breakdowns per minute over the 7 virtual meetings in the *Comapping* project. None of these seriously disrupted the actors' coordination. Interestingly, there were cases of work-process breakdowns that enhanced one actor's heedfulness as he sensed another member of the virtual team acting heedlessly. For example, in the breakdown in Quote 2, the developer's request for repeating an articulation indicates a possible lack of heed. However, the breakdown enhanced the manager's heedfulness as he immediately exploited both audio and visual modalities to alleviate the breakdown. These observations motivate Proposition 2 on managing work-process breakdowns in mediated team coordination.

The contrast between Proposition 2 and the claim that heedless interrelating induces further heedlessness (Weick and Roberts 1993) can be explained through the particular nature of mediated team coordination. The perceived incidents of heedless interrelation in the *Comapping* project would likely not be considered breakdowns in Weick and Roberts's (1993) study. Situated on an aircraft carrier flight deck, their study focused on complex systems of dependencies between many actors with less emphasis on the details of communication between individual actors. As a result, Weick and Roberts (1993) identified incidents in which heedlessness by one actor had ripple effects on the level of heed exhibited by other actors by creating breakdowns in each individual's representation of the social system. In the *Comapping* project, the responding actor exhibited additional heed as he reconstructed his contribution multiple times. In doing so, he relied on the heedful representation of the *Comapping* project that was constructed and shared through the combined use of teleconferencing and real-time collaborative modeling.

> ***Proposition 3 (Organization).*** *Specification, prioritization, and time boxing of tasks through multimodal communication can facilitate collective minding in virtual teams.*

During the early stages of the *Comapping* project, task management had caused significant organizational breakdowns. According to the manager, they initially had a large number of tasks with first priority, making it difficult for the developer to prioritize between them. This experience motivated the structuring of the project into sprints (Rising and Janoff 2000), allowing the team to time box (Jalote et al. 2004) specific tasks. During the mediated virtual meetings, the *Comapping* project negotiated the specification of tasks, their priority, and the time box in which they should be addressed. These interrelated activities were supported by the flexibility of the mindmapping tool, providing a ready-at-hand integration of their negotiations into a shared project model. In this way, the combined use of task specification, prioritization, and time boxing enabled collective minding by helping the three actors contribute and subordinate to the project based on a shared and continuously updated representation of key commitments. These findings motivate Proposition 3 on the organization of mediated team coordination.

To avoid subordination to a system that is envisaged carelessly, a high level of attention should be given to maintaining a heedful representation of the system (Weick and Roberts 1993). In the *Comapping* project, tasks were structured into sprints in the real-time collaborative modeling tool. This time boxing limited the actors' commitment to concurrent tasks and imposed additional structure on their meetings, which reduced the likelihood of information overload. Information overload implies actors have more information available than they can assimilate (Edmunds and Morris 2000) and it leads to a loss of perspective and greater tolerance of error (Eppler and Mengis 2004). Such effects could adversely affect the actors' representation of the system and their ability to interrelate actions heedfully. The adopted organization of the project's coordination created a simple and shared workspace that helped the actors assign and coordinate work dynamically (Dourish and Bellotti 1992).

> ***Proposition 4 (Lifeworld).*** *Collective minding helps virtual team members overcome differences in cultural and professional knowledge across sites without explicitly sharing that knowledge.*

Lifeworld breakdowns occurred when taken-for-granted constitutive knowledge underpinning the coordination of the *Comapping* project was challenged. Interestingly, these breakdowns were relatively rare, accounting for only 18% of incidents, and they mostly related to differences in cultural and professional knowledge across sites. In the lifeworld breakdown in Quote 4, the actors identify differences in cultural knowledge, but immediately move beyond these differences by agreeing on general product requirements. In a different breakdown, the actors explicitly agreed that professional knowledge underlying a specific action didn't need to be shared. These findings motivate Proposition 4 related to lifeworld challenges in virtual team coordination.

Current research has identified serious difficulties related to knowledge sharing across sites in virtual teams (Cramton 2001, Majchrzak et al. 2005, Sole and Edmondson 2002) and coined these the mutual knowledge problem (Cramton 2001) and the situated knowledge challenge (Sole and Edmondson 2002). In response, it has been suggested temporarily to relocate participants physically (Sole and Edmondson 2002), to support the communication of differences in context by information technology (Majchrzak et al. 2005), and to hone the skill of grasping local realities across teams (Cramton 2001). Proposition 4 questions such a strong emphasis on explicitly sharing cultural and professional knowledge across sites as a substitute for everyday sharing of contextual knowledge in collocated teams.

Indeed, the findings from the *Comapping* project and the success of open source development provide a different perspective. In open source development, participants are collectively capable of developing innovative products and services (Ebert 2007, von Hippel 2001, von Hippel and von Krogh 2003) even though they are scattered around the globe with no or little knowledge of local contexts. Knowledge sharing is, of course, a key activity in open source development (Sowe et al. 2008, Spaeth et al. 2008), but the knowledge shared is task-related rather than team-related (Sowe et al. 2008). A core group typically creates the vast majority of new functionality (Koch and Schneider

2002, Mockus et al. 2002); these groups coordinate informally using very simple communication tools like e-mail, listservs, newsgroups, and change management systems, e.g. CVS or Bugzilla (Mockus and Herbsleb 2002). While open source development has specific characteristics, such as developers being experts in a narrowly defined domain and the core group not exceeding a certain size to limit the overhead of informal coordination (Mockus and Herbsleb 2002), these insights corroborate the findings from the *Comapping* project and suggest that mediated team coordination – under certain conditions – can be successful with little or no sharing of knowledge about local contexts.

Turning to the more general literature on team coordination, Proposition 4 is in line with the claim that inadequate individual comprehension can be compensated for through social means (Weick and Roberts 1993). In fact, socio-cognitive theory suggests that high performance teams can be so complex that it is impossible for any single team member to hold all the knowledge required for success (Cannon-Bowers and Salas 2001). In such cases, team members need to specialize and success will therefore depend on the team's ability to coordinate effectively based on the diverse knowledge of several members.

Research Methodology: Multimodal Communication in Virtual Teams

While the use of multimodal communication allows actors to combine different forms of interaction in virtual team meetings, it also leads to unusually complex sets of data. To address this complexity, we separated concerns by drawing on specific distinctions (see Table 3) and by subsequently integrating these into a coherent analytical framework (see Figure 1). Moving from left to right in the presented research framework in Figure 1, our analyses have drawn on speech act theory (Austin 1962, Searle 1969, Winograd 1987) to distinguish between assertive, directive, commissive, declarative, and expressive *articulations*; we have identified assign task, create node, delete node, move node, rename node, prioritize node, and report status as distinct *manipulations* of shared mindmaps; we have drawn on virtual team research (Bjørn and Ngwenyama 2009, Ngwenyama 1998) to distinguish between technology mediation, work process, organization, and lifeworld *communication breakdowns*; and we have analyzed contributions, representation, and subordination as the defining characteristics of *collective minding* (Weick and Roberts 1993).

Drawing on these constructs, we have demonstrated how the combined use of teleconferencing and real-time collaborative modeling allowed the actors in the *Comapping* project to interrelate heedfully and achieve high coordination performance. The evidence is provided through detailed analyses of all the mediated articulations and manipulations between the involved actors and of all the communication breakdowns that occurred during these interactions in the *Comapping* project. These analyses contribute to the literature by going beyond simplistic analyses of individual perceptions as indicators of collective minding (Cooren 2004) and by providing a detailed field study of virtual team coordination focused on micro-level content analysis (Yoo and Kanawattanachai 2001). As a result, they motivate the following proposition:

> ***Proposition 5 (Research approach).*** *Collective minding in virtual teams based on multimodal communication can be investigated through a combination of quantitative and qualitative analyses of articulation acts, manipulation acts, and communication breakdowns.*

Collective minding has previously been investigated through observations and interviews on aircraft carriers (Weick and Roberts 1993); through process documents, interviews, and participation in software requirements' development organizations (Crowston and Kammerer 1998); through questionnaires in student experiments (Yoo and Kanawattanachai 2001); and through conversation analysis of board meeting recordings (Cooren 2004). The framework presented in this paper is similar to Cooren's (2004) conversation analysis, but it differs by mainly relying on one conversation analytic perspective, i.e. language action, rather than also drawing on other perspectives like interactional achievements (Lerner 1992), talking into being (Heritage 1984), and sequence-initiating actions (Schegloff and Sacks 1973). This choice was motivated by our access to audio recordings of the team's teleconferencing as well as video recordings of the team's real-time collaborative modeling; by our access to data from multiple meetings; and by our goal to preserve clarity in detailed analyses of rich data.

Limitations and Implications

Our study has notable limitations regarding case characteristics that call for caution when transferring findings to other contexts. First, many virtual teams are larger and therefore more complex than the *Comapping* project. Increased complexity can make it more difficult for actors to represent the social system and thus more difficult to interrelate their actions heedfully. The size of a virtual team can therefore have a significant influence on collective minding. Second, the national culture of the Russian and Danish participants did not appear to obstruct their ability to coordinate and communicate significantly. Other studies have shown that national diversity across teams may imply difficulties related to communication routines (Maznevski and Chudoba 2000), linguistic differences (Kayworth and Leidner 2000), and weak interpersonal relationships (Kraut et al. 1999). While collective minding helped the *Comapping* project overcome differences in cultural knowledge, it is unclear whether this is transferable to other and more diverse cultural constellations. Third, the technology used in the *Comapping* project was teleconferencing combined with real-time collaborative modeling. The mindmapping tool was developed in the *Comapping* project and the team was therefore highly dedicated to multimodal communication. While the specific technologies are available online for any virtual team to explore, the relationship between virtual team organization and technology may not be equally well aligned in other teams. Finally, the data collection was limited to a specific period of the *Comapping* project. We investigated collective minding at a mature phase of the project, focused on the public release of the product and further investment in the product. Considering previous research into the collective mind in virtual teams (Yoo and Kanawattanachai 2001), we suspect that collective minding may be enacted differently at different project phases, and that these differences may be particularly evident by contrasting project initiation phases with project finalization phases.

As a result, we encourage other researchers to make in-depth field investigations of collective minding across different project phases. Such investigations could increase our understanding of how task characteristics condition collective minding in virtual teams. Further research is also needed into how different communication technologies affect collective minding. Such studies could help us understand how collective minding can be enabled with similar or radically different technologies. Cultural diversity remains a significant topic in virtual team coordination research (Kayworth and Leidner 2000, Kraut et al. 1999, Maznevski and Chudoba 2000), and future research could help us understand better its interaction with and impact on collective minding. Virtual team size also raises interesting questions for future research concerning how the complexity of a virtual team organization influences collective minding. Finally, future research could include additional conversation analytic perspectives – like interactional achievements (Lerner 1992), talking into being (Heritage 1984), and sequence-initiating actions (Schegloff and Sacks 1973) – to broaden this study's unilateral language action perspective (cf. Cooren 2004).

While our study's main emphasis has been on theoretical development, there are also preliminary implications for the practical management of virtual teams. First, the combination of teleconferencing and real-time collaborative modeling can under certain conditions enable collective minding in mediated team coordination. Virtual team practitioners who aspire to achieve high coordination performance should therefore explore how the use of multimodal information technologies can more effectively support coordination across sites. Second, the combination of teleconferencing and real-time collaborative modeling can help actors heedfully sense and respond to work-process breakdowns in virtual team coordination. Virtual team practitioners seeking to achieve collective minding should therefore explore how participants can exploit the combined use of technologies to help sense and respond to work-process breakdowns during virtual meetings. Third, specification, prioritization, and time boxing of tasks in combination with real-time collaborative modeling can enable collective minding in mediated team coordination. Virtual team practitioners should therefore consider ways to organize project tasks that support shared models and straightforward coordination of the project. Finally, collective minding can help team members overcome differences in cultural and professional knowledge across sites without explicitly sharing that knowledge. Virtual team practitioners should therefore critically consider the need for early investments in sharing the cultural and professional knowledge across sites based on the project's ability to develop and maintain collective minding through multimodal communication.

Conclusion

This study has provided an enriched socio-cognitive understanding of how multimodal communication can support virtual team coordination. We have demonstrated how the combined use of teleconferencing and real-time collaborative modeling allow actors in the *Comapping* project to interrelate heedfully and achieve high coordination performance. As a result, we have offered detailed propositions that explain how multimodal communication can facilitate collective minding in virtual teams and positively impact on coordination performance. In addition, we have argued that the impact of multimodal communication on virtual team performance can be investigated through a combination of quantitative and qualitative analyses of articulation acts, manipulation acts, and communication breakdowns.

References

Anderson, A. H., R. McEwan, J. Bal, J. Carletta. 2007. Virtual team meetings: An analysis of communication and context. *Computers in Human Behavior* **23**(5) 2558–2580.

Austin, J. 1962. *How To Do Things With Words.* Harvard University Press, Cambridge, MA.

Benbasat, I., D. K. Goldstein, M. Mead. 1987. The case research strategy in studies of information systems. *MIS Quarterly* **11**(3) 369–386.

Bergiel, B. J., E. B. Bergiel, P. W. Balsmeier. 2008. Nature of virtual teams: A summary of their advantages and disadvantages. *Management Research News* **31**(2) 99–110.

Bijlsma-Frankema, K., B. de Jong, G. van de Bunt. 2008. Heed, a missing link between trust, monitoring and performance in knowledge intensive teams. *The International Journal of Human Resource Management* **19**(1) 19–40.

Bjørn, P., O. Ngwenyama. 2009. Virtual team collaboration: Building shared meaning, resolving breakdowns and creating translucence. *Information Systems Journal* **19**(3) 227–253.

Boland, R. J., R. V. Tenkasi, D. Teeni. 1994. Designing information technology to support distributed cognition. *Organization Science* **5**(3) 456–475.

Cannon-Bowers, J.A., E. Salas. 2001. Reflections on shared cognition. *Journal of Organizational Behavior* **22**(2) 195–202.

Cannon-Bowers, J. A., E. Salas, S. Converse. 1993. Shared mental models in expert team decision making. N. J. Castellan, ed. *Individual and Group Decision Making.* Lawrence Erbaum Associates, Hillside, NJ, 221–243.

Carmel, E., R. Agarwal. 2001. Tactical approaches for alleviating distance in global software development. *IEEE Software* **18**(2) 22–29.

Cooren, F. 2004. The communicative achievement of collective minding: Analysis of board meeting excerpts. *Management Communication Quarterly* **17**(4) 517.

Cooren, F. 2006. Arguments for the in-depth study of organizational interactions: A rejoinder to McPhee, Myers, and Trethewey. *Management Communication Quarterly* **19**(3) 327.

Cramton, C. D. 2001. The mutual knowledge problem and its consequences for dispersed collaboration. *Organization Science* **12**(3) 346.

Crowston, K., E. E. Kammerer. 1998. Coordination and collective mind in software requirements development. *IBM Systems Journal* **37**(2) 227–245.

Curseu, P. L., R. Schalk, I. Wessel. 2008. How do virtual teams process information? A literature review and implications for management. *Journal of Managerial Psychology* **23**(6) 628–652.

Dourish, P., V. Bellotti. 1992. Awareness and coordination in shared workspaces. *ACM Conference on Computer-Supported Cooperative Work*. Toronto, Canada, 107–114.

Dubé, L., D. Robey. 2009. Surviving the paradoxes of virtual teamwork. *Information Systems Journal* **19**(1) 3–30.

Ebert, C. 2007. Open source drives innovation. *IEEE Software* **24**(3) 105–109.

Edmunds, A., A. Morris. 2000. The problem of information overload in business organisations: A review of the literature. *International Journal of Information Management* **20**(1) 17–28.

Ensley, M. D., C. L. Pearce. 2001. Shared cognition in top management teams: Implications for new venture performance. *Journal of Organizational Behavior* **22**(2) 145–160.

Eppler, M. J., J. Mengis. 2004. The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines. *The Information Society* **20**(5) 325–344.

Espinosa, J. A., S. A. Slaughter, R. E. Kraut, J. D. Herbsleb. 2007. Team knowledge and coordination in geographically distributed software development. *Journal of Management Information Systems* **24**(1) 135–169.

Fiore, S. M., E. Salas, H. M. Cuevas, C. A. Bowers. 2003. Distributed coordination space: Toward a theory of distributed team process and performance. *Theoretical Issues in Ergonomics Science* **4**(3/4) 340–364.

Galanter, M. 1989. *Cults.* Oxford University Press, New York.

Galegher, J., R. E. Kraut. 1994. Computer-mediated communication for intellectual teamwork: An experiment in group writing. *Information Systems Research* **5**(2) 110–138.

Garcia, A. C., J. B. Jacobs. 1999. The eyes of the beholder: Understanding the turn-taking system in quasi-synchronous computer-mediated communication. *Research on Language and Social Interaction* **32**(4) 337–367.

Gibson, C. B., J. L. Gibbs. 2006. Unpacking the concept of virtuality: The effects of geographic dispersion, electronic dependence, dynamic structure, and national diversity on team innovation. *Administrative Science Quarterly* **51**(3) 451–495.

Griffin, A., J. R. Hauser. 1996. Integrating R&D and marketing: A review and analysis of the literature. *Journal of Product Innovation Management* **13**(3) 191–215.

Herbsleb, J. D., R. E. Grinter. 1999. Architectures, coordination, and distance: Conway's law and beyond. *IEEE Software* **16**(5) 63–70.

Heritage, J. 1984. *Garfinkel and Ethnomethodology.* Polity Press, Cambridge, UK.

Hertel, G., S. Geister, U. Konradt. 2005. Managing virtual teams: A review of current empirical research. *Human Resource Management Review* **15**(1) 69–95.

Holmes, J. 1984. Modifying illocutionary force. *Journal of Pragmatics* **8**(3) 345–365.

Im, H. G., J. Yates, W. Orlikowski. 2005. Temporal coordination through communication: Using genres in a virtual start-up organization. *Information Technology & People* **18**(2) 89–119.

Jalote, P., A. Palit, P. Kurien, V. T. Peethamber. 2004. Timeboxing: A process model for iterative software development. *Journal of Systems and Software* **70**(1–2) 117–127.

Jarvenpaa, S. L., D. E. Leidner. 1999. Communication and trust in global virtual teams. *Organization Science* **10**(6) 791–815.

Jarvenpaa, S. L., T. R. Shaw, D. S. Staples. 2004. Toward contextualized theories of trust: The role of trust in global virtual teams. *Information Systems Research* **15**(3) 250–267.

Johansson, C., Y. Dittrich, A. Juustila. 1999. Software engineering across boundaries: Student project in distributed collaboration. *IEEE Transactions on Professional Communication* **42**(4) 286–296.

Kanawattanachai, P., Y. Yoo. 2002. Dynamic nature of trust in virtual teams. *The Journal of Strategic Information Systems* **11**(3-4) 187–213.

Kanawattanachai, P., Y. Yoo. 2007. The impact of knowledge coordination on virtual team performance over time. *MIS Quarterly* **31**(4) 783.

Kayworth, T., D. Leidner. 2000. The global virtual manager: A prescription for success. *European Management Journal* **18**(2) 183–194.

Kirkman, B. L., J. E. Mathieu. 2005. The dimensions and antecedents of team virtuality. *Journal of Management* **31**(5) 700–718.

Koch, S., G. Schneider. 2002. Effort, co-operation and co-ordination in an open source software project: GNOME. *Information Systems Journal* **12**(1) 27–42.

Kotlarsky, J., P. C. van Fenema, L. P. Willcocks. 2008. Developing a knowledge-based perspective on coordination: The case of global software projects. *Information & Management* **45**(2) 96–108.

Kraut, R., C. Steinfield, A. Chan, B. Butler, A. Hoag. 1999. Coordination and virtualization: The role of electronic networks and personal relationships. *Organization Science* **10**(6) 722-740.

Lerner, G. H. 1992. Assisted storytelling: Deploying shared knowledge as a practical matter. *Qualitative Sociology* **15**(3) 247–271.

Majchrzak, A., A. Malhotra, R. John. 2005. Perceived individual collaboration know-how development through information technology-enabled contextualization: Evidence from distributed teams. *Information Systems Research* **16**(1) 9.

Majchrzak, A., R. E. Rice, A. Malhotra, N. King, S. Ba. 2000. Technology adaptation: The case of a computer-supported inter-organizational virtual team. *MIS Quarterly* **24**(4) 569–600.

Malone, T. W., K. Crowston. 1991. *Toward an Interdisciplinary Theory of Coordination – Tech. Rep. no. 120*. Center for Coordination Science, Sloan School of Management, Massachusetts Institute of Technology.

Malone, T.W., K. Crowston. 1994. The interdisciplinary study of coordination. *ACM Computing Surveys* **26**(1) 87–119.

Martins, L. L., L. L. Gilson, M. T. Maynard. 2004. Virtual teams: What do we know and where do we go from here? *Journal of Management* **30**(6) 805.

Massey, A. P., M. M. Montoya-Weiss, Y. T. Hung. 2003. Because time matters: Temporal coordination in global virtual project teams. *Journal of Management Information Systems* **19**(4) 129–155.

Maznevski, M. L., K. M. Chudoba. 2000. Bridging space over time: Global virtual team dynamics and effectiveness. *Organization Science* **11**(5) 473–492.

McPhee, R. D., K. K. Myers, A. Trethewey. 2006. On collective mind and conversational analysis: Response to Cooren. *Management Communication Quarterly* **19**(3) 311.

Mingers, J. 2001. Combining IS research methods: Towards a pluralist methodology. *Information Systems Research* **12**(3) 240–259.

Mockus, A., R. T. Fielding, J. D. Herbsleb. 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology* **11**(3) 309–346.

Mockus, A., J. D. Herbsleb. 2002. Why not improve coordination in distributed software development by stealing good ideas from open source? *Meeting Challenges and Surviving Success: The 2nd Workshop on Open Source Software Engineering, held in conjunction with the 24th International Conference on Software Engineering*. Orlando, FL.

Montoya-Weiss, M. M., A. P. Massey, M. Song. 2001. Getting it together: Temporal coordination and conflict management in global virtual teams. *Academy of Management Journal* **44**(6) 1251–1262.

Muhr, T. 2008. *ATLAS.ti – The Knowledge Workbench*. Scientific Software Development, Berlin, Germany.

Ngwenyama, O. K. 1998. Groupware, social action and organizational emergence: On the process dynamics of computer mediated distributed work. *Accounting, Management and Information Technologies* **8**(2–3) 127–146.

Powell, A., G. Piccoli, B. Ives. 2004. Virtual teams: A review of current literature and directions for future research. *DATA BASE for Advances in Information Systems* **35**(1) 6–36.

Ramesh, V., A. R. Dennis. 2002. The object-oriented team: Lessons for virtual teams from global software development. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*. Hawaii.

Rising, L., N. S. Janoff. 2000. The Scrum software development process for small teams. *IEEE Software* **17**(4) 26–32.

Robey, D., H. M. Khoo, C. Powers. 2000. Situated learning in cross-functional virtual teams. *IEEE Transactions on Professional Communication* **43**(1) 51–66.

Rogers, Y., J. Ellis. 1994. Distributed cognition: An alternative framework for analyzing and explaining collaborative working. *Journal of Information Technology* **9**(2) 119–128.

Sakthivel, S. 2005. Virtual workgroups in offshore systems development. *Information and Software Technology* **47**(5) 305–318.

Sarker, S., S. Sahay. 2003. Understanding virtual team development: An interpretive study. *Journal of the Association for Information Systems* **4**(1).

Sarker, S., S. Sahay. 2004. Implications of space and time for distributed work: An interpretive study of US-Norwegian systems development teams. *European Journal of Information Systems* **13**(1) 3–20.

Sbisà, M. 2001. Illocutionary force and degrees of strength in language use. *Journal of Pragmatics* **33**(12) 1791–1814.

Schegloff, E. A., H. Sacks. 1973. Opening up closings. *Semiotica* **7** 289–327.

Schwaber, K., M. Beedle. 2001. *Agile Software Development with Scrum.* Prentice Hall, Upper Saddle River, NJ, USA.

Searle, J. R. 1969. *Speech Acts.* Cambridge University Press, Cambridge, England.

Searle, J. R. 1979. A taxonomy of illocutionary acts. *Expression and Meaning: Studies in the Theory of Speech Acts.* Cambridge University Press, Cambridge, 1–29.

Sherif, K., R. Zmud, G. J. Browne. 2006. Managing peer-to-peer conflicts in disruptive information technology innovations: The case of software reuse. *MIS Quarterly* **30**(1) 339–356.

Singh, B. 1992. *Interconnected Roles (IR): A Coordination Model – Tech, Rep. CT-84-92*. Microelectronics and Computer Technology Corp., Austin, TX.

Sole, D., A. Edmondson. 2002. Situated knowledge and learning in dispersed teams. *British Journal of Management* **13**(2) 17–34.

Sowe, S. K., I. Stamelos, L. Angelis. 2008. Understanding knowledge sharing activities in free/open source software projects: An empirical study. *Journal of Systems and Software* **81**(3) 431–446.

Spaeth, S., S. Haefliger, G. von Krogh, B. Renzl. 2008. Communal resources in open source software development. *Information Research* **13**(1) paper 332.

Starbuck, W. H., F. J. Milliken. 1988. Challenger: Fine-tuning the odds until something breaks. *Journal of Management Studies* **25**(4) 319–340.

Swanson, E. B., N. C. Ramiller. 2004. Innovating mindfully with information technology. *MIS Quarterly* **28**(4) 553–583.

Von Hippel, E. 2001. Innovation by user communities: Learning from open-source software. *MIT Sloan Management Review* **42**(4) 82–86.

Von Hippel, E., G. von Krogh. 2003. Open source software and the "private-collective" innovation model: Issues for organization science. *Organization Science* **14**(2) 209–223.

Warkentin, M. E., L. Sayeed, R. Hightower. 1997. Virtual teams versus face-to-face teams: An exploratory study of a Web-based conference system. *Decision Sciences* **28**(4) 975–996.

Weick, K. E. 1983. Contradictions in a community of scholars: The cohesion-accuracy tradeoff. *Review of Higher Education* **6**(4) 253–267.

Weick, K. E., K. H. Roberts. 1993. Collective mind in organizations – Heedful interrelating on flight decks. *Administrative Science Quarterly* **38**(3) 357–381.

Weick, K. E., K. M. Sutcliffe, D. Obstfeld. 1999. Organizing for high reliability: Processes of collective mindfulness. *Research in Organizational Behavior* **21** 81–124.

Wiesenfeld, B. M., S. Raghuram, R. Garud. 1999. Communication patterns as determinants of organizational identification in a virtual organization. *Organization Science* **10**(6) 777-790.

Winograd, T. 1987. A language/action perspective on the design of cooperative work. *Human–Computer Interaction* **3**(1) 3–30.

Wright, P. C., R. E. Fields, M. D. Harrison. 2000. Analyzing human-computer interaction as distributed cognition: The resources model. *Human–Computer Interaction* **15**(1) 1–41.

Yin, R. K. 2003. *Case Study Research: Design and Methods*. (Third ed.). Sage Publications Inc, Thousand Oaks, California.

Yoo, Y., P. Kanawattanachai. 2001. Developments of transactive memory systems and collective mind in virtual teams. *The International Journal of Organizational Analysis* **9**(2) 187–208.