Co-sourcing in software development offshoring

A case study of risk perception and alleviation

Schlichter, Bjarne Rerup; Persson, John Stouby

Published in:
eProceedings of the 8th International Research Workshop on Information Technology Project Management (IRWITPM)

Publication date:
2013

Document Version
Accepted author manuscript, peer reviewed version

Co-sourcing in software development offshoring: A case study of risk perception and alleviation

Bjarne Rerup Schlichter*
Aarhus University, Denmark
brs@asb.dk

John Stouby Persson*
Aalborg University, Denmark
john@cs.aau.dk

*) The authors contributed equally to the paper

ABSTRACT

Software development projects are increasingly geographical distributed with offshoring, which introduce complex risks that can lead to project failure. Co-sourcing is a highly integrative and cohesive approach, seen successful, to software development offshoring. However, research of how co-sourcing shapes the perception and alleviation of common offshoring risks is limited. We present a case study of how a certified CMMI-level 5 Danish software supplier approaches these risks in offshore co-sourcing. The paper explains how common offshoring risks are perceived and alleviated when adopting the co-sourcing strategy in a mature (CMMI level 5) software development organization. We found that most of the common offshoring risks were perceived and alleviated in accordance with previous research, with the exception of the task distribution risk area. In this case, high task uncertainty, equivocality, and coupling across sites was perceived more as risk alleviation than risk taking. This perception of task distribution was combined with high attention to the closely interrelated structure and technology components in terms of CMMI and the actors' cohesion and integration in terms of Scrum.

Keywords

Risk Management, Co-sourcing, Distributed Software Projects, CMMI, Scrum, Agile

INTRODUCTION

Global competition, need for flexibility and resources with new types of expertise as well as reduction of costs drives software developing companies to engage in geographical distributed software projects (Lacity et al. 2009; Persson et al. 2009). Companies may pursue these opportunities by engaging in co-sourcing, where an outsourcing provider and a client meld their IT competencies to accomplish the clients work (Kaiser and Hawk 2004). Nevertheless, as in other business engagements there are risks associated with this practice. While software companies can improve their business processes by the use of CMMI (capability maturity model, integrated) and agile methods, the risks related to offshoring may still be present. Therefore, more studies on the dynamic interactions between out-sourcing and firm capabilities and emergent models of IT outsourcing are needed for understanding how to manage risks across a portfolio of contracts and suppliers (Lacity et al. 2010). Earlier studies have identified risk factors and alleviation methods as such in traditional supplier-client offshoring relationships (Iacovou and Nakatsu 2008; Lamersdorf et al. 2012; Persson and Mathiassen 2010; Singh and Nigam 2012). However, available research is limited on how co-sourcing shapes the perception of offshoring risks where intermediate organizations are involved. More specifically, there is limited research of how an agile method such a Scrum combined with the highest CMMI level influences the perception and mitigation of risks in software development offshoring. Leading us to the following research question:

How are common offshoring risks perceived and alleviated when adopting the co-sourcing strategy in a mature (CMMI level 5) software development organization?

This paper presents how *Systematic*, a CMMI level 5 company applying Scrum, perceives and alleviates risks in a co-sourcing environment across two countries involving an intermediate organization. First, the paper introduces the theoretical background on co-sourcing and risk alleviation in offshoring. The research approach section describes the case and how we collected and analyzed data. The findings section presents our analysis of the company's alleviation of risk and we identify three risk areas where the case company pursues a high level of risk contrary to

the suggestions from the literature. Hereafter we discuss how our analysis address the research question and contributes to previous research. Finally, we summarize the conclusion of the paper.

THEORETICAL BACKGROUND

Offshore outsourcing involves cross-organizational transactions by the use of external agents to perform one or more organizational activities (Dibbern et al. 2004). In software development, this can apply to everything from the use of contract programmers to third-party facilities management. A software risk denotes an aspect of a development task, process, or environment, which, if ignored, increases the likelihood of project failure (Lyytinen et al. 1998). Both domestic and offshore outsourcing in software development involves risks (Nakatsu and Iacovou 2009) and numerous research efforts have investigated risks particular to offshoring and distribution (Iacovou and Nakatsu 2008; Lamersdorf et al. 2012; Persson and Mathiassen 2010; Singh and Nigam 2012). However, the pursued ideal of an effective offshoring setup may differ significantly among software companies, influencing the perception and prioritization of offshoring risks. While risk frameworks have different organizational attention shaping in terms of Leavitt's (1964) socio-technical model involving structure, actors, technology, and task (Lyytinen et al. 1998), the pursued organizational setup in offshoring may shape the perception of the associated risks and their alleviation. The four components of structure, actors, technology, and task are strongly related such that changes in one component will have planned or unplanned effects on the others (Lyytinen et al. 1998). Thus, more research is needed of how risk attention is shaped by different strategies for the setup of software development offshoring.

Offshoring setups may pursue high levels of cohesion, interdependency, and integration, while other setups pursue high levels of independence and low coupling among sites. In the pursuit of high cohesion, companies may co-locate the software developers (Persson 2013; Šmite et al. 2010) adopt agile methodologies (Jalali and Wohlin 2012; Persson et al. 2012) and strive for virtual team setups with high levels of trust (Siebdrat et al. 2009; Søderberg et al. 2013). In addition to the widespread virtual team conceptualization (Curseu et al. 2008; Ebrahim et al. 2009; Martins et al. 2004; Powell et al. 2004; Schiller and Mandviwalla 2007), the high cohesion approach in software development offshoring has been conceptualized as co-sourcing (Kaiser and Hawk 2004). Kaiser and Hawk (2004) define co-sourcing as an outsourcer and client melding their IT competencies to accomplish the client's work. Based on a case study from the financial industry Kaiser and Hawk (2004) suggest five steps involving engagement, commitment, interchange, co-sourcing, and alignment. The goal of alignment in outsourcing means alignment between the two firms in commitment and values through mutually orientated adaptation of strategy and organization (Kaiser and Hawk 2004). However, such mutual adaption may appear less feasible in offshoring setups involving intermediating organizations or departments, as in the Irish bridge involving two-stage offshoring (Olsson et al. 2008) or offshore middlemen (Mahnke et al. 2008). In this way, available research provides limited explanation of how a high cohesion strategy such as co-sourcing shapes the perception of offshoring risks with intermediating organizations.

The processes of software development have different conceptualizations of the ideal practice at the operational level. One of these is the CMMI for development (CMMI Product Team 2006), which prescribes 5 levels of maturity ranging from initial, managed, defined, and quantitatively managed, to optimizing at level 5. Elevating the CMMI certification at the client organization has been suggested as a best practice in offshoring (Rottman and Lacity 2006). Specifically to close the process gap between client and supplier organizations (Rottman and Lacity 2006) since at one point in time more than half of the firms worldwide that were certified at level 5 were in India (Matloff 2005). However, CMMI has been criticized in relation to offshoring since a level 5 supplier certification provides no guarantee of successful outcomes (Matloff 2005). Interestingly, CMMI has been combined with agile methods even though the two approaches may be contradictory in some aspects (Persson 2010; Santana et al. 2009; Turner and Jain 2002). Such successful combination has been shown in a case with CMMI level 5 and Scrum (Sutherland et al. 2008a). The adoption of agile methods in offshoring has several accounts of success (Persson et al. 2012; Sutherland et al. 2008b) and reflects a high cohesion approach to offshoring. However, the presented research provides limited explanation of how an agile method such as Scrum combined with a high CMMI level shape the perception of offshoring risks.

Software development risks can be managed by numerous approaches, e.g. the eight presented by Keshlaf and Riddle (2011) in their developing effort of a 9th approach for distributed settings. In fact, numerous research efforts has proposed risk frameworks for offshoring and distribution of software development (Iacovou and Nakatsu 2008; Lamersdorf et al. 2012; Persson and Mathiassen 2010; Singh and Nigam 2012). Persson et al. (2009) present a framework that systematically integrates a decade of research on global software, virtual teams, distributed projects,

and outsourcing into 8 risk areas and 35 resolution techniques with mutual links. Each of these 8 risk areas (Table 1) is an abstraction of 3 risk factors.

Leavitt's (1964) Socio-technical components	#	Risk Area	Description of high risk
Task	1	Task Distribution	High task uncertainty, equivocality, and coupling across sites.
	2	Knowledge Management	Inhibited knowledge creation, capture, and creation across sites.
Structure	3	Geographical Distribution	High spatial, temporal and goal distribution among sites.
	4	Collaboration Structure	Breakdowns in collaboration, coordination and process alignment across sites.
Actors	5	Cultural Distribution	Dividing language barriers, work culture and cultural bias across sites.
	6	Stakeholder Relations	Low stakeholder commitment, mutual trust and relationship building across sites.
Technology	7	Communication Infrastructure	Limited personal communication, media support, and teleconference management.
	8	Technology Setup	Poor network capability, tool capability and configuration management.

Table 1 Risk areas for distribution of software development (Persson et al. 2009)

The integrative risk management framework for distributed software development (Persson et al. 2009) explicitly address the four components of Leavitt's (1964) socio-technical model and it is arguably compatible with CMMI (Persson and Mathiassen 2010). However, the framework provides no explanation of how these risk areas (Table 1) are perceived and alleviated when adopting the co-sourcing strategy in a mature (CMMI level 5) software development organization. Thus, in the following, we present our research approach for investigating a CMMI level 5 certified software company employing agile methods and co-sourcing in offshoring with the risk management framework as the analytical lens.

RESEARCH APPROACH

This section presents the case and its related context followed by an explanation of how we collected and analyzed data. The case study approach was in the terms of Cavaye (1996) single case with interpretive use of qualitative data for discovery. This interpretive research approach allowed us to investigate how co-sourcing shape offshoring risks in its organizational and cross-cultural context as socially constructed and thus open to several interpretations by organizational actors but also to us as researchers (Klein and Myers 1999; Walsham 1995; Walsham 2006).

The case

The software company *Systematic* was established in 1985 and have more than 450 employees at offices in Denmark, the United Kingdom, the United States, Australia, Germany, Finland and Sweden. They have four core business areas.. *Systematic* is the largest privately owned Danish software development company and is one of few European companies that has reached and sustained a CMMI level 5 certification since 2005 (Pries-Heje et al. 2008). Especially larger customers often requires a high maturity level. Their later addition of the agile method Scrum in 2006 supposedly enhanced the productivity with a factor two (Sutherland et al. 2008a), even though some research has claimed that CMMI can be in conflict with agile methods as Scrum (Santana et al. 2009; Turner and Jain 2002). Scum is an iterative and incremental development model where planning is concurrent to the development activities and the work is divided into smaller chunks (often weekly) called sprints. Each sprint is planned to be self-contained leading to a new running version on the road to the final software product (Jakobsen and Sutherland 2009). *Systematic* has outsourced system development activities offshore for some years, primarily with a cost-reduction focus, with varying degree of success. In 2010, *Systematic* initiated cooperation with the offshoring intermediary company *Conscensia* and in autumn 2012, they bought 25 % of the company. *Conscensia* is a Danish company established in 2006 selling facilitation of software development offshoring to Ukraine (cities of Lviv and Kiev).

The case study takes it's offset in one of the divisions of *Systematic* following the development of 'one of the main product lines. Software development is done by more than 100 developers in seven groups all divided into one or more teams where each team is staffed by both Danish and Ukrainian developers. We have focused on two teams: Team F (20 persons, 7 in Ukraine) and Team H(35 persons, 10 in Ukraine). The Ukrainian software developers are residing in facilities belonging to the Danish service provider *Conscensia*. *Conscensia* , established in 2006, provides offices including infrastructure, finding and recruitment of competences matching clients' needs in relation to both technical and interpersonal skills as well as other human resource services of software developers in form of local facilitation of the developers (e.g. coaching, cultural training, career advisory and assistance with communication between the teams across countries). At the location in Lviv, *Conscensia* is organized with two delivery managers (A and B) with reference to the Vice President (VP) of Global Delivery and a Chief Operating Officer (COO) with reference to the CEO. The CEO and the VP are situated in Denmark. A local IT department manager, a Recruitment Manager and a Career Advisor, supports the COO. In all, more than 100 developers are situated in the Lviv premises.

The two *Systematic* teams, supported by Delivery Manager A, develop mission critical software, primarily based on .Net and Java. Both teams apply Scrum in their development process and they sit in their own open offices at each location. The teams use Intelli/IDEA as Integrated Development Environment, Rational Team Concert (RTC) to manage source code, and Concurrent Version System (CVS) to manage documentation. Lync facilitates the majority of communication, such as live calls and shared screens. Daily scrum meetings are held for 15 minutes in the morning in dedicated rooms using large screens and laptops showing each other's environments. The teams are organized with a product-manager and headed by a project manager and a one or more scrum-master for each sub-team.

Data collection

The data collection included document studies and individual semi-structured interviews with team members and management from both *Systematic* and *Conscensia*. We initiated the case study with informal meetings with managers in *Systematic* (in Denmark) and *Conscensia* (in Ukraine) in spring 2012. To get an overview of the overall organization, we did exploratory interviews with managers and developers in the early summer 2012 in Lviv. We developed an interview guide based on this explorative phase focused on their offshoring challenges and alleviation strategies. This guide supported our semi-structured interviews in Lviv and Aarhus autumn 2012 and spring 2013. The pilot interviews conducted with managers of *Conscensia* and a couple of software developers brought about several changes to the interview guide such as framing and focusing questions for software professionals. They furthermore provided an understanding of the environment and the challenges faced by the organizations and helped identify additional candidates for interviewing.

We interviewed four members off each team with different roles and nationalities as well as managers from *Conscensia* and *Systematic*. After interviewing the Danish side of the case, we interviewed the Ukrainian side once more to qualify observations and challenge provisional findings. Each interview lasted from 40 to 60 minutes, was recorded, and fully transcribed verbatim. To ensure correct information regarding e.g. use of technology and to maintain good relations with the interviewees the transcriptions was sent for verification. In all, we did 19 interviews combined with informal meetings. In addition to the interviews, we took pictures of the premises (offices and facilities for scrum-meetings) and collected supporting documents such as organograms, sketches of workplaces, presentations, and product descriptions.

Data analysis

We analyzed the interview transcripts and documents to uncover the involved participants' attention to or alleviation of risks related to offshoring. Searching for deviations from established theory by approaching the analysis as a critical dialogue between the theoretical frameworks presented in the background section and our empirical work (Alvesson and Kärreman 2007). To identify incidents, alleviation, or perceptions related to offshoring risks, we searched and coded the transcripts in NVivo (Bazeley 2007). We coded statements pertaining to offshoring risks and grouped them to reveal patterns or other findings. For further triangulation, managers in *Systematic* and *Conscensia* reviewed the analyses, which lead to a few corrections providing alternative interpretations and questioning of findings (Klein and Myers 1999). In the following, we present our findings related to the eight risk areas (table 1) for software development offshoring (Persson et al. 2009).

FINDINGS

This section presents the case company's perception and alleviation of risks. For each of the eight risk areas in table 1, we identified the level of risk and the associated perspective in the case (table 2), primarily from our interviews with management. Furthermore, we identified the main two risk alleviation initiatives for each risk area in table 2.

	#	Risk area	Risk attention	Risk alleviation
Task	1	Task Distribution	Pursue *high risk* by: High task equivocality with very limited specification that gives high task uncertainty at the vendor site combined with high coupling requiring extensive cross-site coordination	• Daily scrum meetings based on video-conferencing and extensive code-reviewing • Well defined processes and division of responsibilities (CMMI-5 certified)
	2	Knowledge Management	Pursue *medium risk* by: Knowledge creation and capture is mainly at the client side with only some knowledge integration across sites	• Partial ownership of intermediary company and focus on staff retention • Training by client domain experts done at vendor side
Structure	3	Geographical Distribution	Pursue *low risk* by: Low distribution with limited time zone differences across only two yet distanced sites that share major goals	• Developers in Ukraine participate in only one team • Site selection and recruitment that lower distribution
	4	Collaboration Structure	Pursue *low risk* by: Recruiting collaborative team members and establishing shared coordination mechanisms and processes across sites	• Danish team lead take part in recruitment • Support by delivery manager from intermediary company and CMMI-5 imposed structure
Actors	5	Cultural Distribution	Pursue *low risk* by: Harmonizing the work culture with English as a shared language and low cultural bias to address the fundamental differences across sites	• Screening and training in cultural awareness of staff by intermediary company • Teambuilding during frequent visits across sites
	6	Stakeholder Relations	Pursue *low/medium risk* by: Recruiting committed participants with a team and client oriented identity that trust the organization and manage integration of new members, while still maintaining some differences between sites	• Surveys twice a week of employee satisfaction and opinions with local budgets for team socialization activities • Sharing of customer and product stories
Technology	7	Communication Infrastructure	Pursue *low risk* by: Strong communication support in terms of interaction media and teleconference management but only some social and personal communication	• Standard use of set up across all teams with optional collaborative tools • Video-conferencing of daily Scrum and urge for additional contact across sites
	8	Technology Setup	Pursue *low risk* by: Setting up reliable network capabilities, compatible tools, and configuration management	• Well defined and stable local infrastructure provided by intermediary company • Software Development tools (IDE, RTC, and CVS) highly integrated into CMMI-5 processes across sites.

Table 2: Risk attention and alleviation in *Systematic's* Co-sourcing activities

Three risk areas (1, 2, and 6 in table 2) call for special interest since the case company pursues a high level of risk, yet they do not perceive it as such, contrary to the suggestions from the literature (Persson et al. 2009). An analysis related to these three areas is presented below in more detail. This analysis was guided by the department manager's emphasis on access to more flexible and lower cost resources than found domestically as a main driver for co-sourcing in *Systematic*.

Task Distribution

Systematic pursues a high-risk strategy for this risk area according to the research (Persson et al. 2009) by intentionally providing very limited specification of development tasks for the Ukrainian site. Requiring extensive cross-site coordination for carrying out development tasks, which according to Persson et al. (2009) should be avoided. The sourcing manager at *Systematic* states: *'When talking about outsourcing you tend to forget what the task is about. It is about a team that produces software together. Then they may sit in different places and talk different languages, but that does not change the basic task of collaborating on making software. We would never write a large requirements specification and through it after someone internally. We would never ask a customer for a large requirements specification and then to stay away. Why should you do that just because it's outsourcing'.* In *Systematic* the managers argue that limited specification of the tasks to be done by Ukrainian developers are beneficial for the process, since it promote local understanding and engagement forcing cross-site team integration by dialogue: *'…their contribution is simply larger…' (Systematic* department manager) and *'…there shall not be more Ukrainians than Danes, all should be integrated into teams, able to fulfill all tasks'* (*Systematic* project manager). It seems that this attitude towards task distribution has led to more engaged Ukrainian software developers leading to a more productive environment: '*..they appreciate getting more responsibilities…and I believe that in the future they will be more engaged in training new colleagues'* (*Systematic* project manager).

Daily scrum meetings by video-conferencing appear to support the management of above mentioned high task uncertainty. This combined with well-defined processes and division of responsibilities as imposed by their CMMI level 5 structure. Thus, they manage the high task uncertainty and equivocality by establishing a high certainty for the process of working with these tasks. Furthermore, they cope with high task coupling by establishing high coordination and collaboration capabilities as reflected in the pursuit of low risk for the risk areas related to this (see risk area 4 and 6 in table 2) and supported by the *Conscensia* Delivery Manager A, who constantly monitors and coaches the working processes.

Knowledge Management

Systematic pursues a medium risk by mainly creating and capturing knowledge at the client side opposed to all sites contributing more equally as suggested by Persson et al. (2009). The sourcing manager at *Systematic* states: '*We must get our domain experts to visit Ukraine, the more the Ukrainian team members knows the better….it matters in the daily small decisions how things works in the large context*'. Furthermore, a *Systematic* project manager argue the limited knowledge integration should be reduced as '*it would be nice with more local domain knowledge…we must improve that*'. The limited creation of knowledge at the Ukrainian side exposes *Systematic* for further risk exacerbation if loosing Ukrainian staff due to their desire to learn more: '*…a small issue related to their career-development, they can't get to know everything…the best of them (can) be lost at the top….we have decided to accept that risk'* (*Systematic* department manager). Thus, the managers at *Systematic* are less coherent in their understanding of the knowledge management risk area and its need for alleviation. Also suggesting that the medium risk exposure on knowledge management is pursued less intentionally compared to task distributed.

Systematic approaches knowledge management risks in several ways. As stated in the interviews, domain experts are sent to Ukraine to train the local staff, but more interesting is how *Systematic* benefits from their partial ownership of the intermediary company. The ownership assures that knowledge, e.g. about processes, not will be lost and can be influenced in-directly at board level. At the same time, the intermediary company assists in staff retention by providing alternative employment and career paths for *Systematic* team members when needed. In this way, they reduce the risk of losing knowledgeable staff. Thus, they alleviate their risks in managing the creation and integration of domain knowledge by strong management and structuring of process knowledge.

Stakeholder Relations

Systematic pursues a low/medium risk by maintaining some differences between sites and the attitude of staff towards colleagues from the other sites, even though they are recruiting team-oriented staff. This difference is especially visible in two areas. First, is trust not directly mirrored: *'…it seems that in Denmark trust has swift nature, where in Ukraine….trust must be earned*' (*Systematic* sourcing manager) and there is a factor two difference of how fast Danish and Ukrainian developers are up to speed and of the sub-teams own understanding of how effective they are. It seems that management in *Systematic* perceives Ukrainian developers as cheap but also slower compared to the more expensive developers placed in Denmark.*Systematic* is assisted by *Conscensia* local cultural training and mediation.

To monitor and be able to react on decreasing levels of trust and satisfaction Systems performs biweekly on-line surveys among staff (both Ukrainians and Danes). One of the issues identified during these surveys and related performance talks was to remember to share customer and product stories with the Ukrainian side as done in the Danish offices. Thus, the *Systematic* managers alleviate stakeholder relations risks by treating Ukrainian and Danish developers as equals while still maintaining differences in trust, identity, and integration.

The analysis of interviews identified three risk areas where *Systematic* pursues medium or high risk that involved different risk attention and alleviation than proposed in the literature. In the following, we discuss how the *Systematic* co-sourcing strategy and its attention shaping of common offshoring risks contributes to previous research presented in the theoretical background section.

DISCUSSION

In the following, we review our analysis of the *Systematic* offshoring case in relation to the research gabs presented in the theoretical background section and our research question: How are common offshoring risks perceived and alleviated when adopting the co-sourcing strategy in a mature (CMMI level 5) software development organization?

The investigated case of co-sourcing show that they perceives and alleviates most risk areas in accordance with previous research of software development offshoring risks (Persson et al. 2009). Task distribution, however, is a notable exception as *Systematic* intentionally and successfully pursue high task uncertainty, equivocality, and coupling. Thus, *Systematic* perceive these risk factors of task distribution more in terms of risk alleviation rather than risk taking. The success of this strategy may be explained by their alleviation of the other risk areas related to structure, actors, and technology, allowing a different perception of task distribution. Thus, the co-sourcing strategy (Kaiser and Hawk 2004) allow high task uncertainty by a strong alignment of structure, actors, and technology (supported by a CMMI maturity level 5). This risk attention shaping analysis complement the Lyytinen et al. (1998) use of Leavitt's (1964) socio-technical model to show how risk areas may be not only be alleviated indirectly by addressing the other three components in the system model but also the perception of a risk area. Elaborating how the co-sourcing strategy for offshoring (Kaiser and Hawk 2004) shapes risk attention in socio-technical terms (Leavitt 1964; Lyytinen et al. 1998). The risk attention shaping of offshoring approaches other than co-sourcing is however still an important avenue for future research.

In the theoretical background section, we argued that available research provides limited explanation of how a high cohesion strategy such as co-sourcing (Kaiser and Hawk 2004) shapes the perception of offshoring risks with intermediating organizations (Mahnke et al. 2008). In our case, the intermediary company *Conscensia* primarily facilitated the alleviation of risks directly related to structure and actors. They mediated independent software developers that *Systematic* would approve and integrate into their teams. This makes *Systematic* capable of using their own sophisticated structure and technology benefiting from *Conscensia´s* capabilities. However, as the co-sourcing approach made the two companies more entangled and mutually dependent *Systematic* acquired 25% ownership of *Conscensia*. This is a way of reducing the mutually dependency risk of co-sourcing, that has been given limited attention by Kaiser and Hawk (2004).

We argued in the theoretical background that an agile method such as Scrum (Sutherland et al. 2008a) combined with a high CMMI level (Rottman and Lacity 2006) might shape the perception of offshoring risks. The high CMMI level appeared to provide risk alleviation related to structure and technology in the risk framework (Table 2). Thus, a high CMMI certification may help offshoring risk alleviation, but not to keep up with a vendor's high certification as claimed by Rottman and Lacity (2006). In the *Systematic* case, the vendor side is independent developers partly organized by an intermediary. CMMI helped in terms of a well-defined and continually improved structure for including these developers. On the other hand, the agile method Scrum appear to have shaped the perception of the

task distribution risk area by embracing uncertainty, equivocality, and coupling as unavoidable. They manage this risk area, not only by daily interaction through video-based standup meetings and frequent code-reviews (table 2), but also through successful risk alleviation of the other risk areas in conjunction with the support of CMMI.

The Persson et al. (2009) framework integrates risks related to software development offshoring in accordance with CMMI (Persson and Mathiassen 2010) and socio-technical terms (Leavitt 1964; Lyytinen et al. 1998). However, findings from the *Systematic* case study show the framework has limited attention to the different framing of task distribution with the introduction of agile methods. The framework does not explicate how the understanding and alleviation of one risk area may alleviate or even exacerbate other risk areas. Yet our case study show how these relations can be very important for understanding risk perception and alleviation as seen with the task distribution risk area. These findings, illustrate the importance of future research on tool and framework support for more explicit management of interrelationships between risk components (El-Masri and Rivard 2012).

CONCLUSION

This paper explained how common offshoring risks are perceived and alleviated when adopting the co-sourcing strategy in a mature (CMMI level 5) software development organization. While most of the common offshoring risks were perceived and alleviated in accordance with previous research, the perception of the task distribution risk area was different from previous research on offshoring risks. In this case study of co-sourcing in a mature (CMMI level 5) software development organization, they perceived high task uncertainty, equivocality, and coupling across sites to be risk alleviation rather than risk taking. The perception of task distribution was furthermore shaped by high attention to the closely interrelated structure, actors, and technology components. While our findings show how co-sourcing may shape the risk perception of task distribution, additional research is still needed of the management of interrelationships between different risk components, the underlying software ecology, and how trust influences the processes.

ACKNOWLEDGMENTS

The research has received support from Center for IT Project Management and Innovation at Aarhus University. We also thank *Conscensia* and *Systematic* for their helpfulness and participation. Ternopil National Economic University has provided a living discursive atmosphere and highly appreciated logistic support during stays in Ukraine. Bjarne Rerup Schlichter is professor by honor at TNEU.

REFERENCES

Alvesson, M., and D. Kärreman. 2007, "Constructing Mystery: Empirical Matters in Theory Development," Academy of Management Review (32:4), pp. 1265-1281.

Bazeley, P. 2007. Qualitative Data Analysis with NVivo, Sage Publications Limited, London.

Cavaye, A. L. M. 1996, "Case Study Research: A Multi-Faceted Research Approach for IS," Information Systems Journal (6:3), pp. 227-242.

CMMI Product Team. 2006. "CMMI for Development, Version 1.2," Tech. Report CMU / SEI-2006-TR-008, Carnegie Mellon Software Engineering Institute.

Curseu, P. L., R. Schalk, and I. Wessel. 2008, "How do Virtual Teams Process Information? A Literature Review and Implications for Management," Journal of Managerial Psychology (23:6), pp. 628-652.

Dibbern, J., T. Goles, R. Hirschheim, and B. Jayatilaka. 2004, "Information Systems Outsourcing: A Survey and Analysis of the Literature," ACM SIGMIS Database (35:4), pp. 6-102.

Ebrahim, N. A., S. Ahmed, and Z. Taha. 2009, "Virtual Teams: A Literature Review," Australian Journal of Basic and Applied Sciences (3:3), pp. 2653-2669.

El-Masri, M., and S. Rivard. 2012. "Towards a Design Theory for Software Project Risk Management Systems." Proceedings of the International Conference on Information Systems.

Iacovou, C. L., and R. Nakatsu. 2008, "A Risk Profile of Offshore-Outsourced Development Projects," Communications of the ACM (51:6), pp. 89-94.

Jakobsen, C. R., and J. Sutherland. 2009. "Scrum and CMMI Going from Good to Great," AGILE '09 Conference, pp. 333-337.

Jalali, S., and C. Wohlin. 2012, "Global Software Engineering and Agile Practices: A Systematic Review," Journal of Software: Evolution and Process (24:6), pp. 643-659.

Kaiser, K. M., and S. Hawk. 2004, "Evolution of Offshore Software Development: From Outsourcing to Cosourcing," MIS Quarterly Executive (3:2), pp. 69-81.

Keshlaf, A., and S. Riddle. 2011, "Web and Distributed Software Development Risks Management: WeDRisk Approach," International Journal on Advances in Software (3:3 and 4), pp. 447-460.

Klein, H. K., and M. D. Myers. 1999, "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems," MIS Quarterly (23:1), pp. 67-93.

Lacity, M. C., S. A. Khan, and L. P. Willcocks. 2009, "A Review of the IT Outsourcing Literature: Insights for Practice," The Journal of Strategic Information Systems (18:3), pp. 130-146.

Lacity, M. C., S. Khan, A. Yan, and L. P. Willcocks. 2010, "A Review of the IT Outsourcing Empirical Literature and Future Research Directions," Journal of Information Technology (25:4), pp. 395-433.

Lamersdorf, A., J. Münch, A. F. Viso Torre, C. R. Sánchez, M. Heinz, and D. Rombach. 2012, "A Rule-Based Model for Customized Risk Identification and Evaluation of Task Assignment Alternatives in Distributed Software Development Projects," Journal of Software Maintenance and Evolution: Research and Practice (24:6), pp. 661–675.

Leavitt, H. J. 1964. "Applied Organization Change in Industry: Structural, Technical and Human Approaches," New Perspectives in Organizational Research, Wiley, Chichester, pp. 55-71.

Lyytinen, K., L. Mathiassen, and J. Ropponen. 1998, "Attention Shaping and Software risk—a Categorical Analysis of Four Classical Risk Management Approaches," Information Systems Research (9:3), pp. 233-255.

Mahnke, V., J. Wareham, and N. Bjorn-Andersen. 2008, "Offshore Middlemen: Transnational Intermediation in Technology Sourcing," Journal of Information Technology (23:1), pp. 18-30.

Martins, L. L., L. L. Gilson, and M. T. Maynard. 2004, "Virtual Teams: What do we Know and Where do we Go from here?" Journal of Management (30:6), pp. 805-835.

Matloff, N. 2005, "Offshoring: What can Go Wrong?" IT Professional (7:4), pp. 39-45.

Nakatsu, R. T., and C. L. Iacovou. 2009, "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), pp. 57-68.

Olsson, H. H., E. O. Conchuir, P. J. Ågerfalk, and B. Fitzgerald. 2008, "Two-Stage Offshoring: An Investigation of the Irish Bridge," MIS Quarterly (32:2), pp. 257-279.

Persson, J. S. 2013. "The Cross-Cultural Knowledge Sharing Challenge: An Investigation of the Co-Location Strategy in Software Development Offshoring," Proceedings of IFIP WG 8.6 International Working Conference on Transfer and Diffusion of IT, June 27-29, pp. 310–325.

———. 2010. Managing Distributed Software Projects, PhD thesis, Department of Computer Science, Aalborg University, Denmark.

Persson, J. S., and L. Mathiassen. 2010, "A Process for Managing Risks in Distributed Teams," IEEE Software (27:1), pp. 20-29.

Persson, J. S., L. Mathiassen, and I. Aaen. 2012, "Agile Distributed Software Development: Enacting Control through Media and Context," Information Systems Journal (22:6), pp. 411–433.

Persson, J. S., L. Mathiassen, J. Boeg, T. S. Madsen, and F. Steinson. 2009, "Managing Risks in Distributed Software Projects: An Integrative Framework," IEEE Transactions on Engineering Management (56:3), pp. 508-532.

Powell, A., G. Piccoli, and B. Ives. 2004, "Virtual Teams: A Review of Current Literature and Directions for Future Research," ACM SIGMIS Database (35:1), pp. 6-36.

Pries-Heje, J., J. Nørbjerg, I. Aaen, and T. Elisberg. 2008, "The Road to High Maturity: How the First Danish Company Reached CMMI Level 5 in 100 Months," PA Nielsen & K.Kautz (2008).Software Processes & Knowledge: Beyond Conventional Software Process Improvement.Aalborg: Software Innovation Publisher, pp. 163-192.

Rottman, J. W., and M. C. Lacity. 2006, "Proven Practices for Effectively Offshoring IT Work," MIT Sloan Management Review (47:3), pp. 56-63.

Santana, C., C. Gusmão, L. Soares, C. Pinheiro, T. Maciel, A. Vasconcelos, and A. Rouiller. 2009. "Agile Software Development and Cmmi: What we do Not Know about Dancing with Elephants," Proceedings of the 10th International Conference, XP2009.pp. 124-129.

Schiller, S. Z., and M. Mandviwalla. 2007, "Virtual Team Research an Analysis of Theory use and a Framework for Theory Appropriation," Small Group Research (38:1), pp. 12-59.

Siebdrat, F., M. Hoegl, and H. Ernst. 2009, "How to Manage Virtual Teams," MIT Sloan Management Review (50:4), pp. 63-68.

Singh, A., and A. R. K. Nigam. 2012, "Risks Identification in an Offshore-Onshore Model Based IT Engagement," International Journal of Computer Applications (48:14), pp. 31-41.

Šmite, D., C. Wohlin, T. Gorschek, and R. Feldt. 2010, "Empirical Evidence in Global Software Engineering: A Systematic Review," Empirical Software Engineering (15:1), pp. 91-118.

Søderberg, A., S. Krishna, and P. Bjørn. 2013, "Global Software Development: Commitment, Trust and Cultural Sensitivity in Strategic Partnerships," Journal of International Management.

Sutherland, J., C. Ruseng Jakobsen, and K. Johnson. 2008a. "Scrum and Cmmi Level 5: The Magic Potion for Code Warriors," Proceedings of the 41st Annual Hawaii International Conference on System Sciences, pp. 466-474.

Sutherland, J., G. Schoonheim, E. Rustenburg, and M. Rijk. 2008b. "Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams," AGILE '08 Conference, pp. 339-344.

Turner, R., and A. Jain. 2002. "Agile Meets CMMI: Culture Clash Or Common Cause?" Extreme Programming and Agile Methods — XP/Agile Universe 2002pp. 153-165.

Walsham, G. 2006, "Doing Interpretive Research," European Journal of Information Systems (15:3), pp. 320-330.

———. 1995, "Interpretive Case Studies in IS Research: Nature and Method," European Journal of Information Systems (4:2), pp. 74-81.