Trust in Co-sourced Software Development

Schlichter, Bjarne Rerup; Persson, John Stouby

Published in:
Proceedings of the Eighth Mediterranean Conference on Information Systems

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

TRUST IN CO-SOURCED SOFTWARE DEVELOPMENT

Complete Research

Bjarne Rerup Schlichter*, Aarhus University, Denmark, brs@badm.au.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. Co-sourcing is a highly integrative and cohesive approach, seen successful, to software development offshoring. However, research of how dynamic aspects of trust are shaped in co-sourcing activities is limited. We present a case study of how the co-sourcing relationship between a certified CMMI-level 5 Danish software company and an offshoring supplier can be conceptualized as an Abstract System. An Abstract System is a dis-embedded social system (such as banking) that is trusted despite lack of detailed understanding or personal trust relations. The paper suggest how certain work practices among developers and managers can be explained using a dynamic trust lens based on Abstract Systems, especially dis- and re-embedding mechanisms.

Keywords: Dynamic Trust, Abstract Systems, Co-sourcing, Distributed Software Projects, CMMI, Scrum, Agile, Offshoring

1 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 issues of trust associated with this practice. While software companies can improve their business processes by the use of CMMI (capability maturity model, integrated (Team, 2010)), the trust issues related to offshoring may still be present. Therefore, more studies on the dynamic interactions between offshoring, firm capabilities, and emergent models of IT outsourcing are needed for understanding how to manage trust in development projects. Earlier studies identified trust factors in traditional supplier-client offshoring relationships (Kelly and Noonan, 2008, Westner and Strahringer, 2010, St John et al., 2013) and dynamic aspects of trust in dedicated information systems implementation projects (Schlichter, 2010a). However, available research is limited on how trust is maintained in a co-sourcing relationship. More specifically, research of how trust is managed in software development offshoring from a company with the highest CMMI level is limited. Leading us to the following research question:

How is trust maintained in co-sourced software development?

This paper presents how *Systematic*, a certified CMMI level 5 software company, perceives trust issues in a co-sourcing environment across two countries involving the offshoring provider *Conscensia*. First, the paper introduces the theoretical background on dynamic trust in distributed settings. The research approach section describes the case, data collection, and data analysis. The findings section presents our analysis of the actors' perceptions of trust to co-sourced software

development as an *Abstract System*. Hereafter we discuss how our analysis address the research question and contributes to previous research. Finally, we summarize the conclusion of the paper.

2 Theoretical Background

This section provides a theoretical and contextualised introduction to offshoring, co-sourcing, and related trust issues.

2.1 Offshoring and co-sourcing

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 transaction can apply to everything from the use of contract programmers to third-party facilities management.

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, available research provides limited explanation of how a high cohesion strategy such as co-sourcing shapes trust in the software development process for offshoring.

The processes of software development have different conceptualizations of the ideal practice at the operational level. One of these is the CMMI for development (Team, 2010), 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 is a suggested best practice in offshoring (Rottman and Lacity, 2006). Specifically to close the process gap between client and supplier organizations (Rottman and Lacity, 2006). At one point in time more than half of the firms worldwide that were certified at level 5 were in India (Matloff, 2005). However, a CMMI level 5 supplier certification provides no guarantee of successful offshoring (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). Our case company show a such successful combination of CMMI level 5 and Scrum (Sutherland et al., 2008a). The adoption of agile methods in offshoring reflects a high cohesion approach to offshoring, that has several accounts of success (Persson et al., 2012, Sutherland et al., 2008b).

Trust is a key construct in outsourcing to an unknown workforce since a 'trusted' partnership is needed. Especially 'techie' to 'techie' relationships promotes trust (Ågerfalk and Fitzgerald, 2008). Even though a vast amount of research in outsourcing and trust is published in the leading Information Systems journals, only few papers directly addresses aspects of outsourcing in an offshore setting. The majority of these papers address trust in relation to contractual relationships between legal entities but not how actors establish trust into a team based software development process. E.g. they research how trust between business partners can be seen as one of the factors that determines how development-outsourcing projects are contracted, managed and priced to address i.e. risk (Gefen et al., 2008).

2.2 Trust

The success and failure of virtual teams (e.g. distributed software development groups communicating mainly through information and communication technologies (ICTs)) is contingent upon trust. This contingence is because trust functions like the glue that holds and links virtual teams together. Virtual teams need to build trust swiftly at the very outset. However, past studies on virtual teams also found that the trust in virtual teams appeared to be fragile. Thus, maintaining trust is as important as to build it (Kanawattanachai and Yoo, 2002). Traditional control mechanisms imported from a face-to-face communication environment are less effective in an ICT-mediated communication environment (Piccoli and Ives, 2003). Virtual teams are a particularly fruitful ground for gaining an understanding of how trust moderates, rather than directly affects, outcomes. Contextualized views of trust in global virtual teams are there for called for (Jarvenpaa et al., 2004). The effects of using trust as a managerial intervention depends on the situation and conditions (Schlichter and Rose, 2013). In situations with weak structure, managers may attempt to change the level of trust. Increases in trust are likely to have a direct, positive impact on a team member's attitudes and perceived outcomes. In situations with moderately strong structure, increases in trust are likely to have contingent impacts through other factors. In situations with strong structure, increases in trust are likely to have little or no effect on work outcomes (Jarvenpaa et al., 2004). Team members' frequent communication in the team provides reassurance that others are attending to the task and increases a member's early trust in the team and feelings of cohesiveness' (Jarvenpaa et al., 2004). Once individuals accumulate sufficient information to assess a team member's trustworthiness, the effects of swift trust declines and knowledge-based trust formed by team members' *behaviours* (perceived ability, integrity, and benevolence) become dominant. The use of ICT increase perceived risk of team failure, which reduce the likelihood that team members engage in future trusting behaviours' (Robert Jr et al., 2009).

Emotional (e.g. trust) issues of outsourcing relations related to software development have not been in focus and thus are in need of further studies (Kelly and Noonan, 2008). However, no framework explains how trust is maintained when co-sourcing from a mature (CMMI level 5) software development organization. Thus, we present a dynamic trust framework (Rose and Schlichter, 2012) as the analytical lens for investigating co-sourced software development from a CMMI level 5 certified software company. Below, we introduce trust issues related to outsourcing, virtual teams and presents the lens of Abstract Systems (Giddens, 1990) to establish a frame for conceptualizing dynamic trust in these settings.

We focus on the managers' trust in the co-sourced software development process (the 'abstract system') to reach their expected goals. Abstract systems are combinations of technical means, procedures, professional expertise and other structures. Examples of this are legal and banking systems those actors approaches and uses despite their limited detailed understanding of how they work. They trust them. This trust in abstract systems enables dynamism in modern societies, by allowing social individuals to act with confidence in the absence of personal knowledge of, or contact with, the structures, people, and actions embodied in the system. As a direct consequence, trust in abstract systems allows e.g. the use of a bank without detailed knowledge of its procedures or established relationships with its employees. Abstract systems are thus disembedding mechanisms, enabling time-space distanciation and providing security and guarantees to their users. An abstract system is a means to stabilize relations across time and space — 'something to trust in' (Walsham, 1998). Trust in abstract systems produces dynamism in society by allowing individuals to proceed in situations of uncertainty, freeing (mental) resources and enabling social interactions across time and space. The absence of such trust forces social actors to take many actions to reduce risk and uncertainty, to control situations by face-to-face interactions and confidence-building measures, and to set in place procedures and regulations to govern social interactions (Giddens, 1990), as rephrased by (Schlichter, 2010b). We adapt the following principle analytical constructs related to trust derived from Giddens by Schlichter & Rose (2013) as structured in figure 1.

- Abstract system – dis-embedded social system trusted despite lack of detailed understanding or personal trust relations
- Trust – in persons and in abstract systems.
- Time-space distanciation – the ability of a social system to function over time and space without the physical co-presence of its social actors, sustained by trust.
- Dis-embedding, re-embedding – processes where an abstract system is removed from immediate close contact, and temporarily made personal again.
- Access point – a point where a lay person makes contact with the abstract system.
- Chronic reflexion – constant evaluation of our social situation and actions (including the trustworthiness of people and abstract systems).
- Ontological security - confidence in the robustness and sustainability of self-identity and belief in the continuity of social practice, sustained by trust in people and abstract systems.

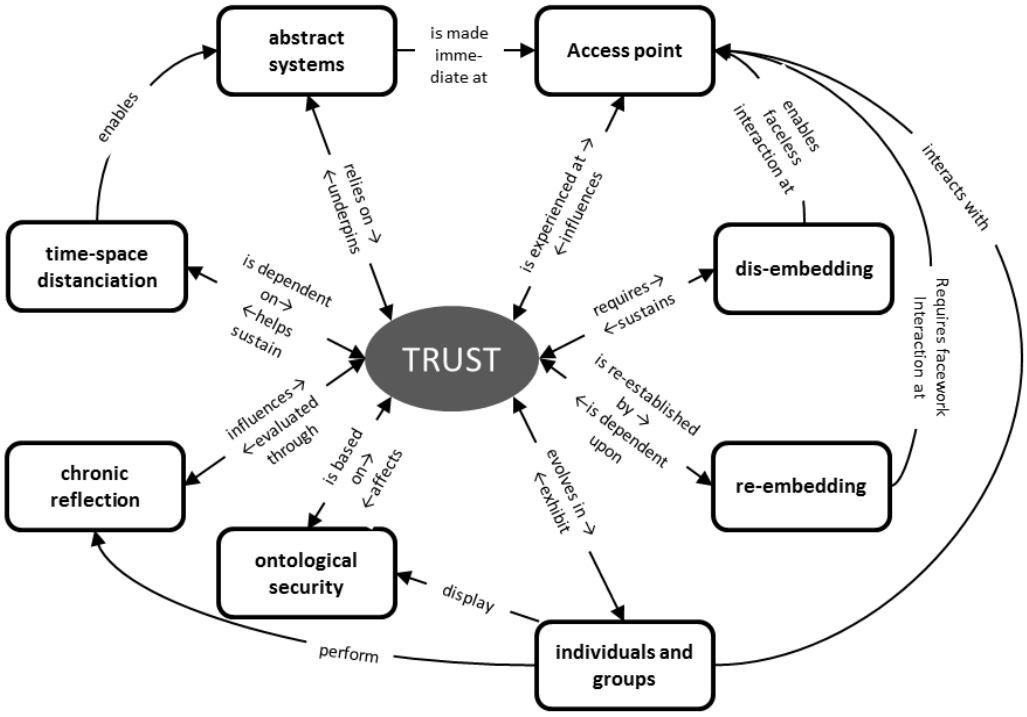

Figure 1: Construct relationships in Giddens' account of trust in abstract systems (Schlichter and Rose, 2013)

We claim that the outsourcing relationship in the present co-sourced development project can be understood as an *Abstract System* and hence that trust can be analyzed using the above-mentioned constructs.

3 Research Approach

This section presents the case and its related context followed by an explanation of how we collected and analysed data. The case study approach is 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 shapes trust in offshoring 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).

3.1 The case

The software company *Systematic*, established in 1985, have more than 450 employees at offices in Denmark, the United Kingdom, the United States, Australia, Germany, Finland, and Sweden. *Systematic* is the largest privately owned Danish software development company and is one of few European companies that since 2005 reached and sustained a CMMI level 5 certification (Pries-Heje et al., 2008). Especially larger public organizations often require 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 claim that CMMI can be in conflict with agile methods as Scrum (Santana et al., 2009, Turner and Jain, 2002). Scrum is an iterative and incremental development approach where planning is concurrent to the development activities and the work is divided into smaller chunks 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 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 its 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 focused on two teams: Team F (20 persons, 7 in Ukraine) and Team H (35 persons, 10 in Ukraine). The Ukrainian software developers reside in facilities belonging to the Danish service provider *Conscensia*. *Conscensia* provides offices including infrastructure, finding, and recruitment of competences matching the clients' needs. Their human resource effort involves both technical and interpersonal skill based recruitment as well as local facilitation of the software developers (e.g. coaching, cultural training, career advisory, and assistance with communication across countries). *Conscensia* has two delivery managers in Lviv (A and B) with reference to the Vice President (VP) of Global Delivery and a Chief Operating Officer (COO) with reference to the chief executive officer (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 scrum-master for each sub-team.

3.2 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 Denmark autumn 2012, spring 2013 and spring 2014. 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. The interviews lasted from 40 to 60 minutes; they were 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. We did 19 interviews combined with informal meetings. In addition to the interviews, we photographed the premises (offices and facilities for scrum-meetings) and collected supporting documents such as organograms, sketches of workplaces, presentations, and product descriptions. The present trust research, focus on eight interviews with four participants from team A. The interview guide (appendix 1) was quite open addressing risks and indirectly areas related to trust, but was not directly based on the trust constructs form figure 1.

3.3 Data analysis

We analysed the interview transcripts and documents to uncover the involved participants' attention to trust related to co-sourced software development. Searching for deviations from established theory by approaching the analysis as a critical dialogue between the theoretical frameworks (the Schlichter & Rose model in figure 1) presented in the background section and our empirical work (Alvesson and Kärreman, 2007). To identify incidents or perceptions related to offshoring and trust, we searched and coded the transcripts in NVivo (Bazeley, 2007). We coded statements pertaining to trust issues 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 based on the model of dynamic trust (Schlichter and Rose, 2013) (figure 1) for co-sourced software development.

4 Findings

In this section, we present the project manager's trust issues related to the co-sourced development process. For each of the eight trust constructs in figure 1, we identified relations of trust and the associated constructs in the case (table 1), primarily from our interviews with management, but also enlightened by software developers in the teams.

#	Trust construct	Frequency	Issues related to the trust-construct
1	Technical means	6	The use of electronically scrum boards Video conferencing – screens and com. Rational Team Concert software
2	Procedures / structures	13	The SCRUM approach to system development CMMI-5 procedures followed Support structure at Conscensia
3	Professional expertise	7	Education of staff Earlier experience from outsourcing
4	Individuals and groups	2	Basically all the actors andgroups : Teams, Projects, Managers, Developers,
5	Time-space distanciation	7	The Co-located team meetings Work on same sprints from different locations.
6	Access point	2	Staffs PC's with tools Scrum-meetings and other project meetings
7	Dis-embedding and re-embedding	30	The use of CMMI-5 Risk alleviation techniques Contact through video-conferences Social gatherings and Country visits
8	Chronic reflection	9	Thoughts on how co-sourcing developed in different settings Learning process at team leads
9	Ontological security	2	The actors firm confidence on the stability of the setup.

Table 1: Abstract Trust Constructs observed in the case

First, we argue that observations from the case confirms that a co-sourced development process can be conceptualized as an Abstract System hence allowing us to apply the analytical framework presented above (Figure 1). An Abstract system is a dis-embedded social system trusted despite lack of detailed understanding or personal trust relations. The social system in consideration for our case study is an abstract system that is developing software. The abstract system is *used* by the management represented by a set of actors: The project manager and the deputy director. These *users* (named 'lay-persons' according to the Abstract System vocabulary) does not have a full and complete understanding on what's going on during the actual software development process but trusts the abstract system which *underpins* it – as well as the abstract system *relies* on this trust. The trust is *dynamic*, influences in both directions. A set of artefacts, i.e. form of technical embodiments (contracts, agreements, IT-based tools) *constitutes* the abstract system which also includes humans (software developers) and well-established procedures (i.e. CMMI-5). In the present conceptualization, managers are the actors approaching the abstract system.

Figure 2 models the dynamic aspects of trust in the Conscensia/Systematic case, as an abstract system of a co-sourced development project. The model illustrates how the actor, the project manager, interacts with the development project through both face work interactions, configuration workshops, and faceless interactions as well as own configuration work using own computer.

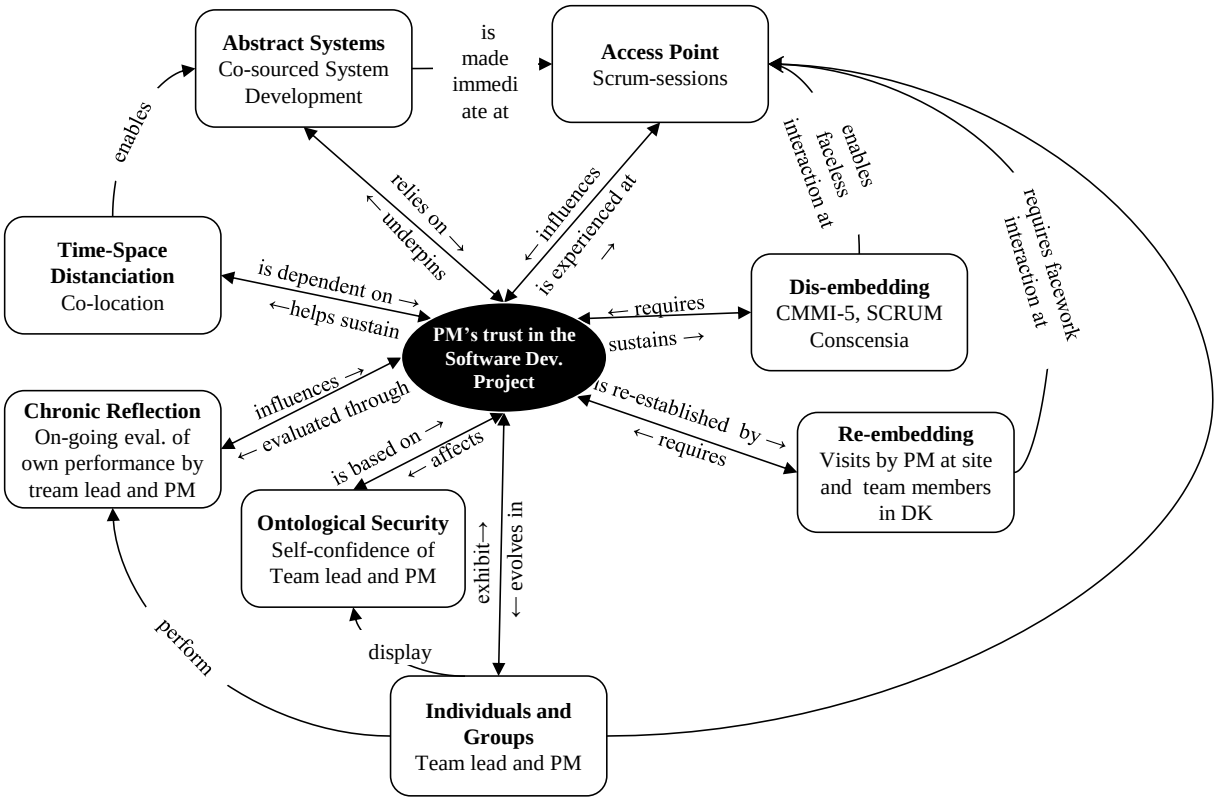

Figure 2: Contextualized model of dynamic trust in co-sourcing

The conceptualisation in Figure 2 show how the generic abstract trust constructs from the case analysis (table 1) can be contextualised.

Two constructs (Dis-embedding and Re-embedding) calls for special interest since the case company pursues a high level of task distribution, and hence a higher level of trust in the co-sourced software development process (Schlichter and Persson, 2013). An analysis related to these two constructs is presented below in more detail. 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*, guided this analysis.

4.1 Dis-embedding

Dis-embedding is the lifting of social relations out of immediate context of personal relations that becomes stabilized across time and space. We observed several indications of dis-embedding in the case, where the abstract system of software development (the 'sprints') executes across the different locations (the 'time-space' distanciation) to the managers satisfaction. As stated by one of the team leads: "*We have an integrated set up so I don't split our teams into a Lviv team and a Aarhus team, we are more like a single team but just located in different locations.*" (Team lead A).

One example of the dis-embedding mechanisms is the structured set of procedures (e.g. the CMMI-5 procedures applied) framing the development process:

"*I have a possibility to compare this with my previous places of work. The company I was working at, it was my first company, I was working there for three years and they also got some CMMI certification but I don't think that processes were used on this company so much as they are on 'Systematic' and I can see that it really just works. We follow the defined process and I think it helps in our daily work, we just know what to do and how to do.*" (Team lead).

Another dis-embedding mechanism is the activities by the offshoring company Conscensia, both in form of cultural difference training: (Software developer); in form of day-to-day on-site support to the development process and local HR processes, e.g.: "*We have some performance talks with <them>, but this is mainly taken care of the <Conscensia>HR department in Ukraine with my input*" (Systematic manager).

Our observations from the case show how the structured development processes and on-site support from the hosting company are dis-embedding mechanisms enabling the actors trust into the software development process.

4.2 Re-embedding

Re-embedding are processes where an abstract system, removed from immediate close contact, temporarily is made personal again. From the interviews, we see two examples of temporary personalisation. The first is the *physical* movement of staff between Systematics premises in Ukraine and Denmark:

"[We go to Denmark sometimes] Yes, I think that our cooperation is going quite well and I think this is one of the main reasons for that is because we meet in person even though LYNC is a great tool it cannot replace live communication. Yes, so we do travel. We have some people from Danish side coming here to Lviv. I think actually in common I'm going three times a year….For one to two weeks." (Team lead).

The team lead state that this face-to-face contact cannot be replaced by on-line communication, even though the use of on-line communication, e.g. in form of the tool LYNC somehow also serves re-embedding purposes by giving the interaction a physical touch:

"*We do that by video-calls. At both locations, we have webcams at the desks so we can see the person we are talking with. Not because we do not know him, but it is always nice to see the face – to see how he reacts when told about a mistake – to see if he becomes crossed about it or still smiles*" (Manager A) and "*one of the most important is Microsoft LYNC which we are using for communication and for our video calls, video meetings, and just for messaging. I think this is one of most important as well*." (Team lead).

Our observations from the case show how trust is re-established through the personal (face-to-face) interactions supported by physical movement and video-calls. These re-embedding mechanisms compensates for the lack of completely dis-embedded structures.

5 Discussion

In this paper, we report an interpretive case study to answer the research question: *How is trust maintained in co-sourced software development?* We found that co-sourced software development can be conceptualized as an Abstract System. Trust in this Abstract System was notably maintained by dis- and re-embedding mechanisms. Dis-embedding was facilitated by the SCRUM and CMMI-5 processes and the structures provided by the offshoring company. Re-embedding was facilitated through physical gatherings (moving staff between Ukraine and Denmark) and videoconferencing. This study show the usefulness of the model of dynamic trust (Schlichter and Rose, 2013) in explaining the maintenance of trust in software development offshoring. Maintenance of trust is a key challenge in offshoring with virtual team setups (Siebdrat et al., 2009, Søderberg et al., 2013) that needs to be understood as a concurrently interpersonal and structural phenomenon. Our findings suggest that trusting a high cohesion strategy to offshoring such as co-sourcing (Kaiser and Hawk, 2004) is not only explained through re-embedding but also dis-embedding by other processes and structures. In the Systematic/Conscenscia case these defined process were Scrum and CMMI-5

(Sutherland et al., 2008a), however other processes may also facilitate dis-embedded trust to co-sourcing.

Previous IS research have to a large extend addressed trust in relation to contractual relationships between legal entities, not how actors establish trust into a team based software development process. E.g. researching how trust between business partners can be seen as one of the factors that determines how development-outsourcing projects are contracted, managed and priced to address i.e. risk (Gefen et al., 2008). Our study provides a contextualized view of trust (see Figure 2) called for in global virtual team research (Jarvenpaa et al., 2004). Based on the model of dynamic trust (Schlichter and Rose, 2013) and the corresponding findings from our case study, we suggest that such a contextualized view of trust can be understood in terms of: Technical means, procedures / structures, professional expertise, individuals and groups, time-space distanciation, access point, Dis- and re-embedding, chronic reflection, and ontological security (Table 1). These abstract trust constructs, may be used in future studies to investigate how trust is maintained in different offshoring contexts for software development.

Trust is an emotional issue in software development offshoring that needs further studies (Kelly and Noonan, 2008). Our contextualized model of dynamic trust in co-sourcing based on a mature (CMMI level 5) software development organization as the case contributes to this call for research. (Kelly and Noonan, 2008) studied trust in offshoring with a very opaque development process. We present how trust in a case with a very well defined development processes is not only dis-embedded but also frequently re-embedded in ways similar to their "relationship work".

Previous research also suggests that the use of ICT increased perceived risk of team failure, which reduced the likelihood that team members would engage in future trusting behaviours (Robert Jr et al., 2009). However, our findings suggest that re-embedding through not only physical gatherings but also videoconferencing is a central component in maintaining trust in a co-sourcing relationship. Thus, our findings support that team members' frequent communication increases trust in the team and feelings of cohesiveness' (Jarvenpaa et al., 2004). However, while Jarvenpaa et al. (2004) argues that situations with strong structure, increases in trust are likely to have little or no effect on work outcomes, management in *Systematic* were highly attentive to maintaining trust in their offshoring relationship with *Conscensian* developers in Ukraine.

6 Conclusion

Based on an interpretive case study, we conclude that the co-sourced software development process can be conceptualized as an Abstract System and that the conceptualization provides insight into the intrinsic dynamic trust relations. We claim that trust is maintained in co-sourced software development by the use of dis- and re-embedding mechanisms:

- The formal process of SCRUM and CMMI-5 and the facilitation management (cultural training / HR support / general process support) done by the offshoring company provides structures and hence a high level of dis-embedding
- Re-embedding is provided through physical gatherings (moving of staff between Ukraine and Denmark) and videoconferencing.

We acknowledges the need for a more in depth analysis and hence description of the presented constructs and their relationships. During the next phase of our research we will refine the model given in figure 2 as well as dig deeper into how specific aspects of the software ecology influences trust during the software development process.

7 Acknowledgments

The research 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 provided a living discursive atmosphere and highly appreciated logistic support during stays in Ukraine. The first author is professor by honour at TNEU.

References:

ALVESSON, M. & KÄRREMAN, D. 2007. Constructing mystery: Empirical matters in theory development. *Academy of Management Review,* 32**,** 1265-1281.

BAZELEY, P. 2007. *Qualitative Data Analysis with NVivo,* London, Sage Publications Ltd.

CURSEU, P. L., SCHALK, R. & WESSEL, I. 2008. How do virtual teams process information? A literature review and implications for management. *Journal of Managerial Psychology,* 23**,** 628-652.

DIBBERN, J., GOLES, T., HIRSCHHEIM, R. & JAYATILAKA, B. 2004. Information systems outsourcing: a survey and analysis of the literature. *ACM SIGMIS Database,* 35**,** 6-102.

EBRAHIM, N. A., AHMED, S. & TAHA, Z. 2009. Virtual Teams: a Literature Review. *Australian Journal of Basic and Applied Sciences,* 3**,** 2653-2669.

GEFEN, D., WYSS, S. & LICHTENSTEIN, Y. 2008. BUSINESS FAMILIARITY AS RISK MITIGATION IN SOFTWARE DEVELOPMENT OUTSOURCING CONTRACTS. *MIS Quarterly,* 32**,** 531-542.

GIDDENS, A. 1990. *The Consequences of Modernity,* Cambridge, MA, Polity Press.

JAKOBSEN, C. R. & SUTHERLAND, J. Scrum and CMMI Going from Good to Great. Agile Conference, 2009. AGILE '09., 24-28 Aug. 2009 2009. 333-337.

JALALI, S. & WOHLIN, C. 2012. Global software engineering and agile practices: a systematic review. *Journal of Software: Evolution and Process,* 24**,** 643-659.

JARVENPAA, S. L., SHAW, T. R. & STAPLES, D. S. 2004. Toward Contextualized Theories of Trust: The Role of Trust in Global Virtual Teams. *Information Systems Research,* 15**,** 250-267.

KAISER, K. M. & HAWK, S. 2004. Evolution of offshore software development: From outsourcing to cosourcing. *MIS Quarterly Executive,* 3**,** 69-81.

KANAWATTANACHAI, P. & YOO, Y. 2002. Dynamic nature of trust in virtual teams. *The Journal of Strategic Information Systems,* 11**,** 187-213.

KELLY, S. & NOONAN, C. 2008. Anxiety and psychological security in offshoring relationships: the role and development of trust as emotional commitment. *Journal of Information Technology,* 23**,** 232-248.

KLEIN, H. K. & MYERS, M. D. 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly,* 23**,** 67-93.

LACITY, M. C., KHAN, S. A. & WILLCOCKS, L. P. 2009. A review of the IT outsourcing literature: Insights for practice. *The Journal of Strategic Information Systems,* 18**,** 130-146.

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.

MATLOFF, N. 2005. Offshoring: What can go wrong? *IT professional,* 7**,** 39-45.

PERSSON, J. S. 2010. *Managing distributed software projects,* Denmark, PhD thesis, Department of Computer Science, Aalborg University.

PERSSON, J. S. The Cross-Cultural Knowledge Sharing Challenge: An Investigation of the Co-location Strategy in Software Development Offshoring. IFIP WG 8.6 International Working Conference on Transfer and Diffusion of IT, 2013 2013 Bangalore, India., 310–325.

PERSSON, J. S., MATHIASSEN, L., BOEG, J., MADSEN, T. S. & STEINSON, F. 2009. Managing risks in distributed software projects: an integrative framework. *Engineering Management, IEEE Transactions on,* 56**,** 508-532.

PERSSON, J. S., MATHIASSEN, L. & AAEN, I. 2012. Agile distributed software development: enacting control through media and context. *Information Systems Journal,* 22**,** 411-433.

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. *SIGMIS Database,* 35**,** 6-36.

PRIES-HEJE, J., NØRBJERG, J., AAEN, I. & ELISBERG, T. 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***,** 163-192.

ROBERT JR, L. P., DENNIS, A. R. & HUNG, Y.-T. C. 2009. Individual Swift Trust and Knowledge-Based Trust in Face-to-Face and Virtual Team Members. *Journal of Management Information Systems,* 26**,** 241-279.

ROSE, J. & SCHLICHTER, B. R. 2012. Decoupling, re-engaging: managing trust relationships in implementation projects. *Information Systems Journal*.

ROTTMAN, J. W. & LACITY, M. C. 2006. Proven practices for effectively offshoring IT work. *MIT Sloan Management Review,* 47**,** 56-63.

SANTANA, C., GUSMÃO, C., SOARES, L., PINHEIRO, C., MACIEL, T., VASCONCELOS, A. & ROUILLER, A. Agile software development and cmmi: What we do not know about dancing with elephants. Agile Processes in Software Engineering and Extreme Programming, 2009. Springer, 124-129.

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.

SCHLICHTER, B. R. 2010a. Development of trust during large scale system implementation. *Journal of Cases on Information Technology,* 12**,** 1-17.

SCHLICHTER, B. R. 2010b. Dynamic Trust in Implementation of Large Information Systems – Conceptualised by Features from Giddens's Theory of Modernity. *Systems, Signs & Actions,* 4**,** 1-22.

SCHLICHTER, B. R. & PERSSON, J. S. 2013. Co-sourcing in software development offshoring: A case study of risk alleviation *8th International Research Workshop on IT Project Management* Milano, Italy.

SCHLICHTER, B. R. & ROSE, J. 2013. Trust dynamics in a large system implementation: six theoretical propositions. *European Journal of Information Systems,* 22**,** 455-474

SIEBDRAT, F., HOEGL, M. & ERNST, H. 2009. How to manage virtual teams. *MIT Sloan Management Review,* 50**,** 63-68.

ŠMITE, D., WOHLIN, C., GORSCHEK, T. & FELDT, R. 2010. Empirical evidence in global software engineering: a systematic review. *Empirical Software Engineering,* 15**,** 91-118.

ST JOHN, J., VEDDER, R. & GUYNES, C. S. 2013. Relationship Changes In IT Offshoring. *International Journal of Management & Information Systems (IJMIS),* 17**,** 131-134.

SUTHERLAND, J., JAKOBSEN, C. R. & JOHNSON, K. Scrum and CMMI Level 5: The Magic Potion for Code Warriors. Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, 7-10 Jan. 2008 2008a. 466-466.

SUTHERLAND, J., SCHOONHEIM, G., RUSTENBURG, E. & RIJK, M. Fully distributed scrum: The secret sauce for hyperproductive offshored development teams. Agile, 2008. AGILE'08. Conference, 2008b. IEEE, 339-344.

SØDERBERG, A.-M., KRISHNA, S. & BJØRN, P. 2013. Global Software Development: Commitment, Trust and Cultural Sensitivity in Strategic Partnerships. *Journal of International Management*.

TEAM, C. P. 2010. CMMI(R) for Development, Version 1.3, Improving Processes for Developing Better Products and Services.: Software Engineering Institute.

TURNER, R. & JAIN, A. Agile meets CMMI: Culture clash or common cause? Extreme Programming and Agile Methods—XP/Agile Universe 2002, 2002 2002. Springer, 153-165.

WALSHAM, G. 1995. The Emergence of Interpretivism in IS Research. *Information Systems Research,* 6**,** 376-395.

WALSHAM, G. 1998. IT and changing professional identity: Micro-studies and macro-theory. *Journal of the American Society for Information Science,* 49**,** 1081-1089.

WALSHAM, G. 2006. Doing interpretive research. *European Journal of Information Systems,* 15**,** 320-330.

WESTNER, M. & STRAHRINGER, S. 2010. Determinants of success in IS offshoring projects: Results from an empirical study of German companies. *Information & management,* 47**,** 291-299.

ÅGERFALK, P. J. & FITZGERALD, B. 2008. OUTSOURCING TO AN UNKNOWN WORKFORCE: EXPLORING OPENSOURCING AS A GLOBAL SOURCING STRATEGY. *MIS Quarterly,* 32**,** 385-409.

Appendix 1: Except of the interviewguide

1) Introduction (to the setting, small talk to assure relaxed situation, since last time)
2) Task Distribution
    a. How are tasks specified?
    b. Who specifies tasks?
    c. What do you do if you don't understand a task?
    d. What do you do if you do not know how to solve a task?

   e. How are tasks divided across sites?
   f. How do you coordinate task dependencies between sites?

3) Trust Issues
   a. What influences your believes in the success full completion of a project?
   b. How can this change?
   c. What is a successful project?
   d. Are there any differences in how you trust different actors in the project?
      i. How / Why ?