

Understanding a Lack of Trust in Global Software Teams: A Multiple-case Study



Research Section

Nils Brede Moe^{1*,†} and Darja Šmite²

¹ SINTEF Information and Communication Technology, NO-7465 Trondheim, Norway

² University of Latvia, LV-1050, Raiņa bulv.19, Rīga, Latvia

Many organizations have turned towards globally distributed software development (GSD) in their quest for cheap, higher-quality software that has a short development cycle. However, this kind of development has often been reported as being problematic and complex to manage. There are indications that trust is a fundamental factor in determining the success or failure of GSD projects. This article studies the key factors that cause a lack of trust and the effect of lacking trust and present data from four projects in which problems with trust were experienced. We found the key factors to be poor socialization and socio-cultural fit, increased monitoring, inconsistency and disparities in work practices, reduction of and unpredictability in communication; and a lack of face-to-face meetings, language skills, conflict handling, and cognitive-based trust. The effect of lacking trust was a decrease in productivity, quality, information exchange and feedback, morale among the employees, and an increase in relationship conflicts. In addition, the employees tended to self-protect, to prioritize individual goals over group goals, and to doubt negative feedback from the manager. Further, the managers increased monitoring, which reduced the level of trust even more. These findings have implications for software development managers and practitioners involved in GSD.

Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: trust; global software development; global software teams; virtual teams; multiple-case study

1. INTRODUCTION

Many organizations have turned towards globally distributed software development (GSD) in their quest for cheap, higher-quality software that has a short development cycle. Nowadays, GSD is becoming the norm (Damian and Moitra 2006). A GSD

team consists of distributed members who collaborate on a common software project while working across geographic, temporal, cultural, political, and organizational boundaries to accomplish an interdependent task (Šmite and Borzovs 2006). Such a working environment presents significant challenges with respect to communication, coordination, and control (Ågerfalk and Fitzgerald 2006). Therefore, GSD is recognized as being considerably more complex to manage than even the most complex in-house project (Karolak 1998, Carmel 1999).

* Correspondence to: Nils Brede Moe, SINTEF ICT, NO-7465 Trondheim, Norway

†E-mail: nils.b.moe@sintef.no



Although a body of knowledge on GSD has been built up, much work remains to be done for it to become a mature discipline: understanding must be deepened, methods and techniques must be developed, and practices must evolve and improve (Sahay *et al.* 2003, Damian and Moitra 2006).

According to Martins *et al.* (2004), GSD teams can be characterized as virtual teams. It is believed that trust is a fundamental factor in determining the success and failure of virtual teams (Grabowski and Roberts 1999, Kanawattanachai and Yoo 2002, Martins *et al.* 2004). Therefore, it is reasonable to believe that trust is also important for GSD teams.

The objective of this article is to understand the importance of trust in GSD by describing the key factors that lead to a lack of trust, and the effect that a lack of trust has on a GSD team. To the best of our knowledge, the existing literature does not address these issues. Several related studies have confirmed this (Edwards and Sridhar 2003, Ali-Babar *et al.* 2006). The core research questions are therefore:

- *What are the key factors that cause a lack of trust in a GSD team?*
- *What is the effect of a lack of trust on the performance of a GSD team?*

The remainder of the article is organized as follows. In Section 2, we first present background information on how work is coordinated in software development, and how this is related to GSD and trust. Then we use the literature to describe the effect of a lack of trust on GSD teams and to identify several factors that work against the development of a high level of trust. In Section 3, we describe our research method in detail. In Section 4, we present the results of our investigation of the role of trust in four projects in a Latvian software company, according to both the findings in the literature and the additional findings from the multiple-case study. We discuss our findings in Section 5. Section 6 concludes and provides recommendations on how to avoid trust-related problems in GSD teamwork.

2. BACKGROUND

2.1. Coordinating Work in Software Development Teams

Software development processes depend significantly on team performance, as does any process that involves human interaction. Two important

factors related to group performance are feedback and communication (Guzzo and Dickson 1996). According to Mintzberg (1989), there are three basic mechanisms that describe the fundamental ways in which work can be coordinated:

1. Mutual adjustment – based on the simple process of informal communication, achieved by a continuous exchange of information among participants;
2. Direct supervision – one person takes responsibility for the work of others by issuing instructions and monitoring their actions;
3. Standardization – of which there are four types: work processes, output, skills (as well as knowledge), and norms.

The coordinating mechanisms may be considered as the most basic elements of structure, the glue that holds the organization together (Mintzberg 1989). Mutual adjustment and direct supervision can be categorized as coordination by feedback (Groth 1999), where coordination is adjusted continually as people observe the effects of their own and others' actions. Standardization can be categorized as coordination by programme (Groth 1999), where coordination is effected through instructions and plans generated beforehand. The mechanisms may act as substitutes for each other to some degree, but all will typically be found in a reasonably well-developed organization.

A software organization often deploys experts in multidisciplinary teams that carry out projects in a complex and dynamic environment. Such organizations can be classified as innovative, where mutual adjustment is the most important coordinating mechanism (Mintzberg 1989). The managers should avoid rigid control (direct supervision), which impairs creativity and spontaneity (Takeuchi and Nonaka 1986). Given that innovation means breaking away from established patterns, innovative organizations should not rely on standardization as the primary mechanism of coordination.

However, given that mutual adjustment in its pure form requires everyone to communicate with everyone, the team or network needs to be compact (Groth 1999).

Nowadays, GSD relies mainly on formal mechanisms (standardization), which rely on detailed architectural design and plans, to address impediments to team communication that result from geographical separation (Ågerfalk and Fitzgerald



2006, Ramesh *et al.* 2006). The limited opportunities for personal contact and constant feedback make it difficult to use mutual adjustment as the principal mechanism for coordination.

2.2. Global Software Development Teams and Trust

Trust is believed to be a fundamental factor for the success or failure of virtual teams (Grabowski and Roberts 1999, Kanawattanachai and Yoo 2002, Martins *et al.* 2004). We define trust as (Mayer *et al.* 1995) follows:

“the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party” (p. 712).

Trust in virtual teams has been studied for a long time, and the effect is well documented. Virtual teams that exhibit a high degree of trust experience the following (Jarvenpaa and Leidner 1999):

- significant social communication;
- predictable communication patterns;
- substantial feedback;
- positive leadership;
- enthusiasm;
- ability to cope with technical uncertainty.

Given that a high degree of trust yields significant social communication, predictable communication and feedback, we conclude that trust is a prerequisite for effective mutual adjustment and is therefore necessary for achieving effective coordination of work, which is important for cooperation and performance (Jarvenpaa *et al.* 2004). Jarvenpaa *et al.* (2004) also argue that an increase in trust in teams that have a weak structure is likely to have a direct positive impact on team members' attitudes and perceived outcomes.

Davidson and Tay (2003) and Vanzin *et al.* (2005) argue that trust is a recurring problem in GSD teams, because of geographical, temporal, organizational, cultural, and political differences among the team members. Carmel (1999) argues that distance is an impediment to building relationships of trust and that dispersed teams meet infrequently or never. Ramesh *et al.* (2006) studied three GSD companies and found that the trust built between the teams

helped to limit the formality with which agreements were specified and thus enabled the development teams to adapt rapidly to the changing needs of the project. Humphrey (1989) argues that when a trusting relationship is assumed, the development process has a new degree of freedom; requirements can be handled more sensibly and a great deal of expensive documentation can be avoided.

2.3. The Effect of Lacking Trust

Using a review of the literature as a basis, Salas *et al.* (2005) argue that mutual trust is needed for the team members to work interdependently. They must be willing to accept a certain amount of risk to rely on each other to meet deadlines, contribute to the team task, and cooperate without subversive intentions. Dirks and Ferrin's (2001) review of the literature on the role of trust in organizational settings demonstrates that trust has either a direct or moderating effect on a variety of variables that pertain to desired performance and behavioural outcomes. In their view, trust facilitates the effects of other determinants on performance or behavioural outcomes because it provides conditions under which certain outcomes are more likely to occur. Bandow (2001) argues that a lack of trust within the group may interfere with how effectively individuals contribute to teams, reduce overall team performance, increase cycle time, create higher costs, and affect product quality.

Several effects of lacking trust have been identified. If one does not trust a partner, it might be difficult to work towards the joint goal and it is likely that the employees will pay more attention to *competitive motives and not to cooperation* (Dirks and Ferrin 2001), and even withdraw from participation because they feel insecure (Bandow 2001). In addition, in a low trust situation, the individuals in a group will direct their efforts towards *individual goals rather than the group's goals* (Dirks and Ferrin 2001), and task conflict within a group is interpreted negatively and subsequently results in *relationship conflict* (Dirks and Ferrin 2001, Salas *et al.* 2005).

If individuals do not trust their manager, they find it difficult to behave as expected; and the management's request is likely to exert a much weaker effect on their behaviour, because they divert resources to *self-protection* (Dirks and Ferrin 2001). In addition, when a manager in whom



employees have little trust gives negative feedback, it is likely that the employees will *doubt the accuracy of the feedback* (Dirks and Ferrin 2001, Salas *et al.* 2005). This will hinder the team leader from managing the team effectively.

A low level of trust is also associated with suspicion about information, and therefore low trust will result in *reduced information exchange and feedback* (Bandow 2001, Dirks and Ferrin 2001, Salas *et al.* 2005).

In a low-trust situation, there will be a fall-off in the mutual monitoring of performance. Such monitoring facilitates common understandings of the team environment and the accurate monitoring of team members' performance. Mutual performance monitoring is thus essential for identifying mistakes and lapses in other team members' actions, and for providing feedback regarding actions on the part of team members that facilitates *self-correction* (Salas *et al.* 2005). A fall-off in mutual performance monitoring will affect all these factors negatively. It will also affect the ability to anticipate other team members' needs through accurate knowledge of their responsibilities (back-up behaviour). This includes the ability to *shift workloads among members* to achieve balance during periods of high workload or pressure (Carmel 1999, Salas *et al.* 2005).

Given that a lack of trust affects team performance negatively (Bandow 2001, Dirks and Ferrin 2001, Salas *et al.* 2005), *productivity and quality* will also suffer.

2.4. Key Factors Causing Lack of Trust

The means by which trust is established in a GSD team differ from how it is established in an internal software development team, where participants often know each other already and are aware that their future personal relationships will reach beyond the current project. Trust in virtual teams needs to be developed quickly because teams may only interact for a short period of time or may be working on a task that is very important and urgent (Jarvenpaa and Leidner 1999, Kanawattanachai and Yoo 2002). Earlier work on trust in the virtual environment has found that short-lived teams do develop high trust, but that they do so by following a swift trust model rather than the traditional model of trust development (Jarvenpaa *et al.* 1998, Jarvenpaa and Leidner 1999). Members of such teams do not have the time to develop trust in a

gradual and cumulative fashion. Rather, the team members act as if trust is present from the start. 'Swift trust' enables members to take action, and this action will help the team to maintain trust and deal with uncertainty, ambiguity, and vulnerability while working on complex interdependent tasks with strangers in a situation of great time pressure (Jarvenpaa *et al.* 1998).

Several factors that affect the level of trust in a virtual team have been identified. In order for team members to maintain/strengthen trust in virtual teams, it is important for them to *socialize* (Jarvenpaa and Leidner 1999, Kanawattanachai and Yoo 2002). Therefore, team members should travel to remote sites to engage in a team-building activity to maintain/strengthen trust (Rocco 1998).

For both developing and repairing trust in virtual teams, *face-to-face meetings* are considered irreplaceable (Carmel 1999, Piccoli and Ives 2003, Bhat *et al.* 2006). Face-to-face contact is the richest communication channel we have, and any electronic channel is significantly poorer (Groth 1999). Hence, if there is no face-to-face communication in a virtual team, this tends to hinder effective communication. For example, when team members communicate about mutual responsibility and obligations, different perceptions of their commitments may develop. This creates a potential for trust to decline (Piccoli and Ives 2003). In fact, Karolak considers lack of trust as a natural consequence of losing face-to-face interaction (Karolak 1998).

Virtual teams in a low-trust situation need *frequent and predictable communication* if trust is to grow. Frequent communication is important for providing constant confirmation that team members are still there and still working (Jarvenpaa *et al.* 2004). If feedback is provided on a regular basis, communication improves, which in turn leads to greater trust and improved team performance (Jarvenpaa *et al.* 1998, Jarvenpaa and Leidner 1999). Team members may experience anxiety, or their level of trust may decline, due to negative interpretations of silence or delays associated with time differences (Piccoli and Ives 2003).

Behavioural controls, such as having members file weekly reports and assigning specific tasks, are associated with a decline in trust (Piccoli and Ives 2003). In addition, too much communication might cause team members to be suspicious that others are monitoring them and this causes a decline in trust (Jarvenpaa *et al.* 2004).



Virtual teams need to focus on cognitive-based trust (e.g. competence, reliability, and professionalism) (Kanawattanachai and Yoo 2002), which is also known as performance-based trust (Sabherwal 1999). It is therefore important to provide task-relevant background information to team members so that they can quickly develop *cognitive-based trust*. If the remote team does not deliver what is expected, cognitive-based trust will decline.

Conflicts in virtual teams, such as global development teams, are inevitable (Karolak 1998) and it is often difficult to maintain trust when conflicts among team members emerge. That being so, an absence of mechanisms for *handling conflict* is a threat to building and maintaining trust (Jarvenpaa and Leidner 1999, Kanawattanachai and Yoo 2002).

Culture influences the common understanding between teammates due to diversity in people's assumptions, behaviours, expectations about leadership practices, team norms, attitudes towards hierarchy, sense of time, and communication styles (Duarte and Snyder 2001, Herbsleb and Moitra 2001). Lack of familiarity with *cultural diversity* is seen as a barrier to achieving trust (Ali-Babar *et al.* 2006).

3. RESEARCH METHOD

The goal of the research reported herein was to understand the importance of trust in GSD; hence, it is important to understand the software practitioners' perceptions of the importance of trust in this context. Therefore, we collected data from supply teams that participated in four globally distributed software projects run by LatSoftware (the company name has been changed for reasons of confidentiality). The projects were chosen because they all reported problems with trust. Hence, we focused on investigating the role of trust between teams in LatSoftware that worked closely with teams from different companies outside Latvia, and between teams in LatSoftware with different locations.

We report on a multiple-case holistic study (Yin 2003), in which we studied one phenomenon in several projects in one company. In a multiple-case study, each case must be selected carefully so that it either (a) predicts similar results or (b) predicts contrasting results but for predictable reasons (Yin 2003). We chose option (a) and picked

four global software development projects that all reported problems with trust. Therefore, the results from each case should not be regarded as objects for comparison; rather, they should be seen as complementary findings that work together to enrich our understanding of a lack of trust in global software teams.

A multiple-case study is considered to yield more robust and compelling evidence than evidence gathered through a single case. In addition, it is considered that conclusions from multiple cases can be generalized to a greater extent than findings from a single case (Yin 2003).

3.1. Study Context

The context for the research was the Latvian software development company LatSoftware, situated in Riga. Latvia has become one of the major centres for outsourcing in Europe (Minevich and Richter 2005). LatSoftware was established in the late 1980s and has been oriented towards the international market, focusing on providing software outsourcing services. LatSoftware has successfully completed more than 200 projects in Latvia, Western Europe, and Scandinavia. At the time of the study, the company got over 380 employees. We believe that LatSoftware is a representative example of typical outsourcing partners from Latvia, because they have been involved in GSD for several years.

While LatSoftware was extending its operation into global markets, quality certification was given a high priority. A new system for managing quality has been implemented and the company has been certified according to the ISO 9001:2000 standards. The quality management system consists of descriptions of processes and procedures for software development, methodological guidelines, templates and forms (for example project plans, contracts, and requirements specifications), job instructions and administrative reports. There are more than 500 documents and around 100 process descriptions. The system's use is mandatory for all projects.

3.2. Data Sources

We used multiple data sources (see Table 1): qualitative interviews with project participants, results from postmortem meetings (Birk *et al.* 2002), and



Table 1. Data sources

Pro name	Duration	Project type	Team location ^a and Team size	Effort	Data collection
A	1995–present	SW product development and maintenance	DE(3) + LV(5)	46080 hours	Interviewed current project manager, previous project manager and one developer Problem checklists
B	2002–2006	SW product development	UK(13) + LV(16)	40480 hours	Interviewed project manager and three team leaders. Postmortem analysis Problem checklists
C	2006	SW pilot product development	SE + LV(3)	320 hours	Interview with project manager Problem checklists
D	2005	SW product development	NO(2) + LV(6) + LV(5)	11680 hours	Interview with project manager, system analysts and testing team manager from Riga Group interview with the remote team from Rezekne Postmortem analysis Problem checklists

^a DE, Germany; LV, Latvia; UK, the United Kingdom; SE, Sweden; NO, Norway.

project problem reports. We interviewed project managers from Latvia because they were able to provide a broad perspective on issues of collaboration, acting as communication hubs between the supplier and customer teams. In addition, projects A and D allowed us to conduct interviews with other project members. Talking to project members without a project manager’s involvement provided a richer view of the projects. Owing to the limited availability of team members, it was not possible to interview project members from projects B and C. For projects B and D, we also used results from post-mortem meetings (Birk *et al.* 2002). A postmortem meeting involves every team member. It focuses first on describing what went well and what did not work in the project, then on conducting a root-cause analysis of the main issues that are identified. Using postmortem meetings, it was possible to identify the root causes of problems related to trust. Post-mortem meetings for projects A and C were not possible, because project managers did not accept spending more resources on the project. The minutes from the postmortem meeting was sent to the participants for approval. Project problems were also recorded using a problem checklist, which made it possible to compare findings between the projects.

We only present data collected from the Latvian teams.

3.3. Data Analysis

For the analysis, we relied mainly on qualitative interviews, because these provide a rich picture of the reasons for, and effects of, a lack of trust. We analysed the data in several steps. First, we read all interviews and postmortem analysis data, and coded the material according to open coding (Strauss and Corbin 1998). During open coding, every passage of the interview was studied to determine exactly what was said and each passage related to trust was labelled with an adequate code. Then we compared fragments from different interviews, but from the same project, that were assigned the same code (axial coding). In axial coding, indicators and characteristics for each concept are searched for, in order to define that concept (Strauss and Corbin 1998).

After the axial coding, we assigned the concepts to the categories of ‘reasons for lacking trust’ and ‘effect of lacking trust’ found in the literature on GSD, teams and virtual teams. We also found some new concepts that were not described in the theory.



For example, we coded: 'Because I did not trust them fully I decided to monitor them more'. This was coded as 'increased monitoring': an effect of lacking trust. To avoid bias and misunderstanding, the conclusions from our coding were sent back to the interviewees for approval.

3.4. Threats to Validity

We now discuss briefly the main threats to construct validity, internal and external validity, and reliability (Yin 2003).

To increase construct validity, multiple sources of evidence were used and the draft case study report was reviewed by key informants. For project C, we collected data from only the project leader (through an interview) and the problem checklist. This was because of the limited access to the other participants in project C. However, we believe that the project manager was able to give a complete and detailed picture of this relatively small project (only 320 hours). To support findings regarding quality and productivity, we also collected measurement data. However, there was no company standard for measuring these data, so they are reported unevenly between the projects.

A threat to the internal validity of our study is the potential bias with respect to selecting the projects that participated in the study. If the selected group is not representative of the population studied, internal validity is threatened. To meet this threat, we selected all the GSD projects that had reported problems with trust and that were available at the start of our study.

The threat to external validity is addressed in this multiple-case study by using literal replication. The four projects represent a replication for understanding what leads to a lack of trust, and the effect of lacking trust, in a GSD team. However, the study is limited because it is only validated from the viewpoint of LatSoftware developers, not LatSoftware partners. The participants only came from Latvia; hence, the findings cannot be generalized widely to practitioners from other countries. Further, the projects only involved teams from Europe, which means that the effect of temporal and political diversity on trust could not be observed.

As for reliability, we have described in detail how the study was performed.

4. RESULTS

We describe four global projects run in the investigated software house, describing why trust was lacking, and the effects of this on each project.

4.1. Project A

4.1.1. Overview

Project A was a long-term ongoing software enhancement project with close collaboration between five Latvian developers and three representatives from a German company that was developing a software product for a customer.

4.1.2. Reasons for a Lack of Trust

The project used modern collaboration tools, such as video conferencing and instant messaging, extensively. However, the Riga team claimed that this did not compensate for the necessity of *face-to-face meetings*.

The *lack of language skills* caused time delays, because some of the developers needed to get e-mails translated when written in German. This resulted in *too little, and unpredictability in, communication*, because it took time to translate and answer e-mail. The lack of language skills also resulted in a loss of quality in the information exchanged, because there is always some information loss after translation.

There were *misperceptions*. The Riga team claimed that their German partners perceived the Riga team as not fully dedicated to the project and therefore tried to control them by constantly *monitoring* their performance. This situation persisted for 10 years, during which time the German team never visited their Latvian colleagues and there was *no possibility for socialization*. When the two teams finally met, it became evident that the German team did not know much about their Latvian partner; they were surprised to see the modern offices with high-level security and technical equipment. Their perception of the remote team members changed and there was a period when the German and Latvian teams met frequently. Overall project performance, and team morale and psychological comfort consequently improved.

However, after the period of improvement, new problems arose. Germans presented high-level ideas with frequent changes without detailed documentation. These received a poor response



from the Latvian side, because the corporate culture of LatSoftware did not support an agile approach. As a result of not meeting deadlines, the Germans began to lose their *cognitive-based trust*. The German manager frequently required LatSoftware to change the project leader because he did not trust the leaders' abilities to add value to the project. *Socio-cultural diversity* that existed between the remote partners only sharpened the disputes and conflicts that arose over time.

4.1.3. Effect of Lacking Trust

The lack of trust generated a negative feedback loop. First, the Riga team experienced a radically *reduced ability to self-correct* and started to *doubt negative feedback* from the remote partner. This initiated *extensive monitoring* from the contracting partner, which in turn affected the level of trust negatively. There was a *decrease in information exchange and feedback*, which resulted in the Latvian colleagues often claiming that they were not even being informed about the important changes in project documents.

When the German partner started to search for more beneficial collaboration partners, the Riga team was placed in an environment of *competition, rather than collaboration*. The resultant project atmosphere *influenced team morale and productivity negatively*, caused *relationship conflicts*, and was the reason for *individual goals dominating over group goals*. Effort overruns of 50–80% per iteration were reported.

4.2. Project B

4.2.1. Overview

Project B was a software product development and enhancement project run by a UK software house that outsourced part of the programming to LatSoftware. The management in the UK made a strategic decision to involve LatSoftware. However, the Latvian project manager felt that the UK team representatives who were directly involved in the collaboration did not like this.

4.2.2. Reasons for a Lack of Trust

The main problem in this project was a *missing consensus on how to cooperate*, because of a *disparity in work practices and poor cultural fit*. For example, the Latvian team experienced inconsistency in the development process regarding requirement

and testing. The UK partner seemed to develop requirements according to one method, but tested the software according to other methods. The Latvian project manager also felt that their partner was unwilling to inform the remote team about changes and deviations in the project plans. The Latvian partner felt that their UK counterpart was *monitoring* them, rather than working collaboratively.

Another important issue was related to *communication and language skills*. The desire to cooperate from the Latvian side met with total indifference from the other side. Such problems as a dominant use of asynchronous communication tools, and unwillingness or slowness on the part of the UK team to respond to suggestions from the Latvian partner, led to *poor and unpredictable communication*. *Poor language skills* among the Latvian team members also resulted in a need to translate the written communication, which delayed communication.

There was also a problem with *cognitive-based trust*. Performance problems within the new development environment resulted in unavoidable time delays. The UK partner perceived this as low productivity by the LatSoftware team, which reduced the cognitive-based trust.

Owing to a *lack of joint conflict handling, poor socialization, and lack of face-to-face meetings*, it was difficult to solve the project problems and to increase and maintain the level of trust.

4.2.3. Effect of Lacking Trust

The Latvian team came to *doubt negative feedback* from the continuously indifferent UK partner, and a generally poor work atmosphere ensued. This *reduced information exchange and feedback* causing *poor collaboration and poor morale*. In addition, the *monitoring increased* in the form of an unreasonable amount of reporting. These effects prevented the effective utilization of resources, which caused periods with a heavy workload with corresponding overtime and stand-by periods with no work. This reduced morale and the motivation to give the client value for money, which again, in the opinion of the Latvian team, *reduced productivity*.

4.3. Project C

4.3.1. Overview

Project C was a pilot project to evaluate the investigated Riga software house as an external provider of coding for a software house in Sweden, which has



recently switched to GSD and outsourcing. Their cooperation started by developing a small piece of software, but the project was later suspended.

4.3.2. *Reasons for a Lack of Trust*

The most important reasons for a lack of trust were *socio-cultural and organizational differences, disparity in work practices, and a lack of common procedures and tools.*

The Swedish partner wanted to start by working in small increments to check whether their Latvian partner could deliver as expected. However, due to the factors just mentioned, both partners faced an increasing complexity of distributed multiteam management. After joint risk-management meetings with the Swedish client, the project manager from Latvia reported that the client needed to change its structure and work practices in order to adjust to collaborative software development projects. However, the Swedish partner realized that they were not ready for such changes in their own organization, even though this would make GSD easier.

The level of trust was also affected by *too little communication, a missing belief in joint performance, and a lack of socialization and face-to-face meetings.* There was also *no conflict handling.*

4.3.3. *Effect of Lacking Trust*

According to the Riga project manager, the client's employees felt insecure about their jobs, due to the corporate decision to switch to GSD and outsourcing. This missing trust in the GSD project *undermined the morale* of the Swedish employees, and could be why the Swedish team acted as *competitors instead of collaborators*, thus causing a *decrease in productivity* for the whole GSD project. Consequently, the client team's individual goals dominated over shared project goals.

All task conflicts within the joint team were interpreted negatively. Loss of trust and unresolved conflicts between the teams finally led to a 50% effort overrun and *suspension of the collaboration.*

4.4. Project D

4.4.1. *Overview*

Project D was a complex project that involved a customer from Norway, a direct supplier from Riga, and a remote team of programmers from Rezekne, a small Latvian town situated in the poorest region,

around 250 km from Riga. The project manager in Riga coordinated both of the LatSoftware teams' activities. We focused primarily on collaboration between the two separated teams in Latvia.

4.4.2. *Reasons for a Lack of Trust*

The main reason for a lack of trust in this project was *problems with technology and communication between the teams.* Despite the fact that the Latvian teams worked for the same company, the Rezekne team, which worked in a poorer region of Latvia, had significant problems with technology and lines of communication. This poor infrastructure increased the number of hours required for compiling code, which resulted in the Rezekne team not delivering as fast as was expected by the Riga team. This was perceived as low productivity by the Riga team, which resulted in cognitive-based trust being eroded. Owing to *too little communication*, the project manager in Riga was not informed about the problems with technology and their consequences before the end of the project. The problem with cognitive-based trust was exacerbated because there was little, or a slow, response from the Rezekne team to project requests from the Riga team. Other important reasons for a lack of trust were the *absence of face-to-face meetings and increased monitoring.* Experiencing significant delays in the project, and little and unpredictable communication, the Riga project manager increased the monitoring to find out what was going on in the remote team, and why they did not deliver as quickly as expected. Instead of trying to solve the problems by arranging face-to-face meetings, the Riga project manager sent a representative to investigate the situation in Rezekne. This increased monitoring confirmed the suspicion on the part of the Rezekne team that they were not trusted by the project manager in Riga.

Lack of socialization, conflict handling and face-to-face meetings made it very difficult to improve trust and solve problems throughout the project. Despite the fact that both teams are situated in the same country, they also experienced *socio-cultural diversity*, which also affected the level of trust negatively.

4.4.3. *Effect of Lacking Trust*

The lack of trust in this project increased the Riga project manager's *desire to control* the remote team. As a result of his attempts at control, the remote



team became *self-protective* and *apprehensive about the Riga manager's feedback*. They also *communicated less with the Riga manager, prioritized their own goals over the group goals*, and experienced *poor morale*. The missing trust, desire to *self-protect*, and *insufficient information exchange* resulted in an *unwillingness to shift the workload* between the two distributed teams, even when problems arose. The amount of unresolved project issues increased and this resulted in both Latvian teams being *poorly motivated to self-correct*, as well as a *fall-off in productivity and quality*. There was a 12% effort overrun, and there were serious problems with the quality. The number of bugs uncovered during system (1144) and acceptance testing (220) were considered significant and were more than expected.

4.5. The Effects of, and Key Factors Causing, a Lack of Trust in the Projects

In addition to the effect of lacking trust found in the literature, we found three further effects by analysing our data:

- *Increased monitoring (projects A, B and D)* – when they lack trust in supplier performance and experience a lack of direct control due to geographic and temporal distribution, managers struggle with a desire to control, instead of cooperating with, the remote teams. It seems that some software managers often start with at least a subconscious fear that the developers cannot be trusted to perform without close supervision and control. This results in increased monitoring (Humphrey 1989) and extra time is needed for reporting. A negative feedback loop ensues, because increased monitoring results in a lack of trust, and a lack of trust leads to increased monitoring.
- *Undermined morale of the employees (all projects)* – a lack of trust creates a negative atmosphere that results in the psychological discomfort of the members.
- *Threat of project cancellation (projects A and C)* – we also found that a lack of trust may put the overall collaboration in jeopardy.

In Table 2, we present the key effects of lacking trust that were found in the study

In addition to key factors that cause a lack of trust found in the literature, we found two further factors by analysing our data:

Table 2. The main effects of lacking trust found in the projects

The main effects of lacking trust	Projects			
	A	B	C	D
Competition and non-cooperation	✓		✓	
Individual goals over group goals	✓		✓	✓
Relationship conflict	✓		✓	
Implementation of self-protective measures	✓			✓
Doubting the veracity of negative feedback from manager	✓	✓		✓
Reduction in information exchange and feedback	✓	✓		✓
Team fails to self-correct	✓			✓
Workload not shifted among team members.				✓
Decrease in productivity and quality	✓	✓	✓	✓
Increased monitoring	✓	✓		✓
Undermined morale of the employees	✓	✓	✓	✓
Threat of project cancellation	✓		✓	

- *Lack of language skills (projects A and B)* – leads to poor socialization and communication problems, because employees with poor language skills need to get e-mails and documentation translated, and tend to be afraid to speak over the telephone. A lack of language skills may also result in delays when written documentation needs to be translated.
- *Disparities in work practices* – may lead to a loss of cognitive-based trust, create misunderstandings, make it difficult to cooperate, and increase monitoring.

In Table 3, we present the key factors that caused a lack of trust in the four projects that were studied.

Table 3. Key factors causing lack of trust found in the projects

Reason for lacking trust	Projects			
	A	B	C	D
Poor socialization	✓	✓	✓	✓
Missing face-to-face meetings	✓	✓	✓	✓
Too little communication	✓	✓	✓	✓
Unpredictability in communication	✓	✓		✓
Increased monitoring	✓	✓		✓
Lack of cognitive-based trust	✓	✓		✓
No conflict handling		✓	✓	✓
Poor socio-cultural fit	✓	✓	✓	✓
Lack of language skills	✓	✓		
Disparities in work practices	✓	✓	✓	



5. DISCUSSION

We have described the effects of, and reasons for, a lack of trust, and presented a study on this topic. We now discuss our findings.

5.1. Effects of Lacking Trust

The projects were chosen because they all reported problems with trust in their GSD contexts. Like Bandow (2001), Dirks and Ferrin (2001) we found that when virtual teams lack trust, there may be significant problems with the performance and behaviour of the team members. All the projects reported a decrease in productivity and quality, and that team morale was undermined, as a result of an absence of trust. This confirms the importance of trust for overall project performance. In projects A, B, and D, we also found that there was a decrease in information exchange and feedback because of a lack of trust. These findings were probably related to team members doubting negative feedback from their manager. When this occurs, it is also likely they will not respond to the negative feedback. Teams that have problems with communication and negative feedback find it difficult to self-correct.

An absence of trust is the reason that individual goals appear to be more important than group goals, which reduces the desire to cooperate and increases the desire to self-protect. We found that the managers often responded to a fall-off in cooperation by increasing the monitoring of the remote team, because they felt that they were losing control and needed to find out what was really going on in the remote team. This increased monitoring caused the level of trust to fall even more, resulting in a negative feedback loop. Some managers used the threat of cancelling the project and introducing competitors in an attempt to induce the remote team to work harder. The Germans did this in project A, but contrary to the expected result, morale was undermined even further. The productivity and quality did not increase and perhaps even decreased because of the threat of cancellation.

Only team D had problems with shifting their workload. There could be two reasons for this finding: (a) there were no problems in shifting the workload among the other teams, or (b) shifting the workload was never considered in the other projects. Several teams reported big time delays in their project. This situation could probably have

been improved by allowing teams to shift the workload, but this never occurred probably due to the problems with information exchange and feedback, self-protection, competition, and lack of cooperation.

Our findings demonstrate that a lack of trust can be devastating for a GSD project and that the ensuing problems both create new ones and exacerbate those that already exist.

5.2. Key Factors Causing Lacking Trust

From our study we found that poor socialization, lack of face-to-face meetings, too little communication, and poor socio-cultural fit were reported by all the projects. Lack of face-to-face meetings and poor socialization are probably related, because it is difficult to socialize if you meet seldom or never. The lack of both face-to-face meetings and poor socialization reduced the level of communication and made it unpredictable (projects A, B, and D). On the basis of our results, we believe that the problem with poor socio-cultural fit may have been exacerbated by a lack of face-to-face interaction and too little socialization. An example, of this is the German partner in project A, who waited 10 years before they visited their Latvian partner and then were surprised because they had assumed that the Latvians worked in old-fashioned offices and had a far lower level of security and technical equipment than the Germans.

A lack of language skills (projects A and B) resulted in communication problems, because most of the written documentation and e-mails needed to be translated. We also found that employees with poor language skills tend to be afraid of speaking over the telephone. This also made it more difficult to socialize.

Infrequent meetings, if there are any at all, and primarily asynchronous communication are stumbling blocks for dispersed teams. This can be explained by companies' intentions to reduce their costs by all means available (Carmel 1999). The paradox is that most of the clients expect global teams to be cheap and fast, despite the fact that their methods of communicating make the teams slower (Carmel 1999). Misinterpreting the reasons behind low productivity creates another negative loop and results in a decrease in cognitive-based trust.

Inconsistency in work practices (projects A, B, and C) may reduce cognitive-based trust because



partners solve tasks in a manner different from that which was expected, which results in misunderstandings. Disparities in work practices and too little communication lead to increased monitoring, because the outsourcing partner wants to find out about and control what is going on when proximity is lacking and they are not fully informed of the project's status. A lack of conflict handling (reported by projects B, C, and D) and too little communication also made it difficult to improve the overall level of trust.

LatSoftware is certified according to the ISO 9001:2000 standards and have received several national quality awards. Such certification and recognition helps to generate what has been termed initial trust by projecting credibility, convincing clients to trust in one's capabilities (Ali-Babar *et al.* 2006). However, the effect of initial trust will soon vanish if the projects do not deliver the expected software on time. In project C, the Swedish partner selected their GSD partner because of high initial trust (certificates and quality awards). However, when the results did not meet expectations, the project was cancelled.

Traditional development models, such as the quality management system used by LatSoftware, are often referred to as plan-driven (Boehm and Turner 2003). Such models are rooted in the rationalistic paradigm, which promotes a product-line approach to software development using a standardized, controllable, and predictable software engineering process (Dybå 2000). From this perspective, the standardization of work processes is more important than mutual adjustment when coordinating work in such an organization (Mintzberg 1989). Following the plans and specification were more important than people talking to each other from the perspective of LatSoftware, and this made it difficult to be flexible and respond to new ideas, as shown by the failure to meet the expectations of the German partner in project A. We argue that the focus on plan-driven development model over mutual adjustment is a further reason for lacking trust. Fortunately, LatSoftware changed their work practices and started to work in a more agile way by basing their communication on mutual adjustment. This affected trust positively and the Germans stopped looking for other suppliers. Team performance also improved and the effort overruns were reduced.

5.3. Recommendations

The reason for the failure of global projects is not the lack of capability, but a lack of awareness of the issues, problems, and barriers associated with global work (DeLone *et al.* 2005). There are several ways in which a high level of trust can be improved and maintained in global teams. On the basis of our study, we recommend that a GSD team should discuss the 'key factors' and 'main effects' of lacking trust at an early stage in the collaboration, identify actions to meet the potential problems and also consider the following measures:

- Invest in several face-to face meetings (Karaklak 1998, Carmel 1999, Bandow 2001, Bhat *et al.* 2006, Piccoli and Ives 2003). To increase cognitive-based trust, it is important to present the technical standard of the remote partner, such as technical equipment, offices, and security routines (Ali-Babar *et al.* 2006). Socialization activities for the whole team are also important (Rocco 1998, Jarvenpaa and Leidner 1999, Kanawattanachai and Yoo 2002). This will improve cultural understanding and make it easier to create personal relationships, which are a prerequisite for using mutual adjustment as a coordinating mechanism.
- Communicate expectations early and establish initial rules for handling conflicts. This could be in the form of a contract or trust structure (Bandow 2001).
- Ensure that the team possesses the expected development competence and language skills. This is important for high cognitive trust and better communication. The team should be motivated for the project, and no one should fear losing their job as a result of the cooperation.
- Consider a software development method that provides both flexibility and adaptability, and that uses frequent communication to make it possible to coordinate the work by constant feedback. An iterative method will make it possible to demonstrate a completed portion of the system several times over (Sabherwal 1999), and this will strengthen cognitive-based trust. In addition, trust will grow because the team members will be constantly aware of the status of the project (Humphrey 1989). The desire for obsessive monitoring will thereby decrease. One possible way of implementing the above recommendations is to use agile methods



(Ågerfalk and Fitzgerald 2006). However, due to the physical separation of development teams in GSD, many of the key concepts within agile development (Dybå and Dingsøy 2008), such as pair-programming, face-to-face interaction, and onsite customers, are difficult to implement. Therefore, the software development method should probably balance both the agile and plan-driven approach (Ramesh *et al.* 2006). Make sure that you have a common understanding of the common work process. This can be achieved by holding a process workshop (Dingsøy *et al.* 2004).

- Invest in groupware packages and team intranet to provide effective means of communication and to compensate for the lack of personal contact during the project (Karolak 1998, Carmel 1999). However, be aware that this will not replace the value of face-to-face meetings.

6. CONCLUSION AND FUTURE WORK

GSD has grown substantially richer as a discipline over the past decade. However, for it to mature, much work remains to be done: understanding must be deepened, methods and techniques must be developed, and practices must evolve and improve (Sahay *et al.* 2003, Damian and Moitra 2006). Trust is a recurring problem in GSD teams, because of geographical, temporal, organizational, cultural, and political differences among the team members. Face-to-face meetings, active communication, and socialization, which are commonly used for building trust in in-house software teams, are a hard recipe to follow for global teams, because cost-saving strategies reduce the opportunity to meet.

Drawing upon the related literature, and by studying four projects that were distributed between different locations and that had problems concerning trust, we were able to identify the key factors that cause a lack of trust as follows: poor socialization and socio-cultural fit, lack of face-to-face meetings and language skills, absence of conflict handling and lack of cognitive-based trust, increased monitoring, inconsistency in work practices, and both decrease and unpredictability in communication.

A lack of trust resulted in a decrease in productivity, quality, information exchange, feedback, and morale among the employees, and an increase in

relationship conflicts. In addition, the employees tended to self-protect, prioritize individual goals over group goals, and doubt negative feedback from the manager. The subteams also competed instead of cooperating, and did not shift workload or self-correct. The loss of trust resulted in the managers increasing monitoring and threatening to cancel the project, which reduced the level of trust even more.

These and other findings lead to the conclusion that a company should consider the pros and cons of cross-border collaboration and never start a distributed collaboration unprepared. It is not easy for globally distributed teams to work effectively. An awareness of the importance of trust, the reasons for a lack of trust, and the effect of lacking trust will help to avoid many problems of joint collaboration. However, it is not a simple matter to achieve a high level of trust in GSD teams.

Accordingly, further work should focus on investigating which methods can be applied for building and maintaining trust in GSD.

ACKNOWLEDGEMENTS

We appreciate the input received from project managers and other members of the investigated projects in LatSoftware. We thank Chris Wright for proofreading and Torgeir Dingsøy and Tore Dybå for valuable feedback.

This research is supported by the Research Council of Norway under Grant 181658/I30, the European Social Fund and the Latvian Council of Science within project Nr. 02.2002.

REFERENCES

- Ågerfalk PJ, Fitzgerald B. 2006. Flexible and distributed software processes: old petunias in new bowls? *Communications of the ACM* **49**(10): 26–34.
- Ali-Babar M, Verner JM, Nguyen PT. 2006. Establishing and maintaining trust in software outsourcing relationships: an empirical investigation. *Journal of Systems and Software* **80**(9): 1438–1449.
- Bandow D. 2001. Time to create sound teamwork. *Journal for Quality and Participation* **24**(2): 41.
- Bhat JM, Gupta M, Murthy SN. 2006. Overcoming requirements engineering challenges: lessons from offshore outsourcing. *IEEE Software* **23**(5): 38–44.



- Birk A, Dingsøy T, Stålhane T. 2002. Postmortem: never leave a project without it. *IEEE Software, Special Issue on Knowledge Management in Software Engineering* **19**(3): 43–45.
- Boehm BW, Turner R. 2003. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison Wesley: Boston.
- Carmel E. 1999. *Global Software Teams: Collaborating Across Borders and Time Zones*. Prentice-Hall: Upper Saddle River, NJ, USA.
- Damian D, Moitra D. 2006. Global software development: How far have we come? *IEEE Software* **23**(5): 17–19.
- Davidson EJ, Tay ASM. 2003. *Studying teamwork in global IT support*.
- DeLone W, Alberto Espinosa J, Lee G, Carmel E. 2005. Bridging global boundaries for IS project success. *38th Hawaii International Conference on System Science*. Big Island, HI, United States: Institute of Electrical and Electronics Engineers Computer Society: Piscataway, NJ; 08855–1331.
- Dingsøy T, Moe NB, Dybå T, Conradi R. 2004. A workshop-oriented approach for defining electronic process guides – A case study. In *Software Process Modelling*, Acuña ST, Juristo N (eds). Kluwer Academic Publishers: Boston, MA; 187–205.
- Dirks KT, Ferrin DL. 2001. The role of trust in organizational settings. *Organization Science* **12**(4): 450–467.
- Duarte DL, Snyder LT. 2001. In *Mastering Virtual Teams: Strategies, Tools, and Techniques that Succeed*, 2nd edn, A.W. Company (ed.). Jossey-Bass: San Francisco.
- Dybå T. 2000. Improvisation in small software organizations. *IEEE Software* **17**(5): 82–87.
- Dyba T, Dingsøy T. 2008. Empirical studies of Agile Software Development: A Systematic Review, Information and Software Technology, DOI: 10.1016/j.infsof.2008.01.006.
- Edwards HK, Sridhar V. 2003. *Analysis of the effectiveness of global virtual teams in software engineering projects*.
- Grabowski M, Roberts KH. 1999. Risk mitigation in virtual organizations. *Organization Science* **10**(6): 704–721.
- Groth L. 1999. *Future Organizational Design: The Scope for the IT-based Enterprise*. John Wiley and Sons: New York.
- Guzzo RA, Dickson MW. 1996. Teams in organizations: recent research on performance and effectiveness. *Annual Review of Psychology* **47**: 307–338.
- Herbsleb JD, Moitra D. 2001. Global software development. *IEEE Software* **18**(2): 16–20.
- Humphrey WS. 1989. *Managing the Software Process*. Addison-Wesley: Reading, MA, XVIII, 494 s.
- Jarvenpaa SL, Leidner DE. 1999. Communication and trust in global virtual teams. *Organization Science* **10**(6): 791–815.
- Jarvenpaa SL, Knoll K, Leidner DE. 1998. Is anybody out there? Antecedents of trust in global virtual teams. *Journal of Management Information Systems* **14**(4): 29–64.
- Jarvenpaa SL, Shaw TR, Staples DS. 2004. Toward contextualized theories of trust: the role of trust in global virtual teams. *Information Systems Research* **15**(3): 250–267.
- Kanawattanachai P, Yoo Y. 2002. Dynamic nature of trust in virtual teams. *Journal of Strategic Information Systems* **11**(3–4): 187–213.
- Karolak DWJ. 1998. *Global Software Development*. Wiley: United States – IEEE Computer Society; 172.
- Martins LL, Gilson LL, Maynard MT. 2004. Virtual teams: what do we know and where do we go from here? *Journal of Management* **30**(6): 805–835.
- Mayer RC, Davis JH, Schoorman FD. 1995. An integrative model of organizational trust. *Academy of Management Review* **20**(3): 709.
- Minevich M, Richter F-J. 2005. *Global Outsourcing Report 2005*. Going Global Ventures and HORASIS: New York, Geneva.
- Mintzberg H. 1989. *Mintzberg on Management: Inside Our Strange World of Organizations*.
- Piccoli G, Ives B. 2003. Trust and the unintended effects of behavior control in virtual teams. *Mis Quarterly* **27**(3): 365–395.
- Ramesh B, Cao L, Mohan K, Xu P. 2006. Can distributed software development be agile? *Communications of the ACM* **49**(10): 41–46.
- Rocco E. 1998. *Trust Breaks Down in Electronic Contexts but Can be Repaired by Some Initial Face-to-face Contact*. ACM: Los Angeles, CA, New York.
- Sabherwal R. 1999. The role of trust in outsourced IS development projects. *Communications of the ACM* **42**(2): 80–86.
- Sahay S, Nicholson B, Krishna S. 2003. *Global IT Outsourcing: Software Development Across Borders*. Cambridge University Press: Cambridge.



- Salas E, Sims DE, Burke CS. 2005. Is there a “big five” in teamwork? *Small Group Research* **36**(5): 555–599.
- Smite D, Borzovs J. 2006. A framework for overcoming supplier related threats in global projects. *European Software Process Improvement (EuroSPI), Proceedings*. Springer Verlag: Joensuu, Finland.
- Strauss A, Corbin J. 1998. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications: Thousand Oaks, CA.
- Takeuchi H, Nonaka I. 1986. The new product development game. *Harvard Business Review* **64**: 137–146.
- Vanzin M, Ribeiro MB, Prikladnicki R, Ceccato I, Antunes D. 2005. *Global Software Processes Definition in a Distributed Environment*. United States: Institute of Electrical and Electronics Engineers Computer Society: Greenbelt, MD, Piscataway, NJ; 08855–1331.
- Yin RK. 2003. *Case Study Research: Design and Methods*, 3rd edn, Vol. 5. Sage Publications: Thousand Oaks, CA.