Building Trust in Global Inter-Organizational Software Development Projects: Problems and Practices

Jarkko Pyysiäinen
Helsinki University of Technology
Software Business and Engineering Institute
P.O.B. 9600, FIN-02015 HUT, Finland
Jarkko.Pyysiainen@hut.fi

Abstract

This paper explores problems and potential practices for trust building in global inter-organizational software development networks. The concept and traditional sources of trust are briefly reviewed, and the special problems on trust building in networks are analyzed on the basis of the theoretical framework. Our empirical findings from nine global software development networks show that such networks are facing problems because the traditional sources of trust do not exist in networked conditions. In such networks, trust may emerge occasionally, but maintaining it seems especially challenging. Consequently, building and maintaining trust in globally and organizationally dispersed networks seems to require supportive practices that compensate the deficient sources of trust. On the basis of our empirical data, some successful practices in trust building are outlined.

1. Introduction

Global inter-organizational networks have become increasingly popular in software development [3]. Such networks may include, e.g., several subcontractors or partners working concurrently with customers across distances and relying primarily on communication technologies instead of face-to-face meetings. Work across teams and companies is rather interdependent than independent and the need to orchestrate the work across the whole network is often great. This kind of coordination of work in networks creates new challenges and demands: coordination is simply not possible based on such traditional features as direct face-to-face feedback, common experiences, similarity of backgrounds and co-located decision making [4]. Instead, alternative ways for facilitating cooperation and communication must be utilized. In the field of organizational behavior a growing attention has been paid to the role of trust in such processes: a growing body of literature demonstrates the important benefits of trust for organizations and their members [1,5,8]. Even more importantly, trust is seen as a necessary element in facilitating the functioning of networked organizations [4,7,11] and global software outsourcing relationships [2,9].

While on the one hand trust building seems to be a promising mechanism for overcoming many difficulties related to global software development, it may on the other hand be precisely the virtual and global contexts that constrain the development of trust between companies and teams [e.g. 4]. Lack of face-to-face interaction and informal communication seem especially troubling.

The aim of this paper is to first present a theoretically motivated empirical analysis of the problems encountered in trust building in nine Finnish, global and inter-organizational software development networks. Second, on the basis of the data successful practices in overcoming these problems in trust building are outlined.

2. Theoretical framework

2.1. Concept and sources of trust

For the purposes of this study the concept of trust may be best understood as a relationship between parties, not as a property of an individual: trust prevails if each of the interacting parties acknowledges the right of the other parties to assess the competence and the intentions of their acts [11]. Trust prevails, if parties after this kind of assessment are willing to be vulnerable to – or cooperate with – each other based on the belief that the other is competent, open, concerned and reliable [8]. In this sense trustworthiness of an individual may be a source of trust for others, but trust is realized only in action. Without shared experiences trust is not likely to survive.

This type of assessment of trust between parties also requires a dialogical, negotiating mode of communication. In addition, grounds for this kind of trusting orientation are likely to develop in organizational contexts,

where work is evaluated and control located at the level of the joint project, not at the level of individual contributions. [11] When these antecedents are evaluated from the perspective of global software development networks, it seems that shared experiences at the network level may be rare, but that a negotiating mode of communication and evaluation of work at the level of a joint project are characteristic to some networks.

According to Kramer [5] traditional sources of trust within organizations can be summarized as follows:

Dispositional trust: Predispositions to trust or distrust others tends to be correlated with other personal orientations and styles. Certain features in the behavior of others become associated with stable expectations and it is possible to extrapolate from earlier trust-related experiences.

History-based trust: Trust thickens or thins as a function of cumulative interaction. Individuals' judgments about others' trustworthiness are partly anchored on a priori expectations about others' behavior, and these expectations change as subsequent experience either validates or discredits them. Reciprocity in exchange relations enhances trust while violation of reciprocity erodes it.

Third parties as distributors: Third parties are important because of their ability to diffuse trust-relevant information via informal communication and gossip. On the basis of this kind of mediated information it becomes possible to transfer expectations of existing embedded relationships to newly formed ones.

Category-based trust: Common categories function as vehicles for perceiving common identities and common goals. A shared membership in a salient category (woman, researcher) can provide basis for presumptive trust and a sense of familiarity. Membership in a category is associated with a tendency to attribute positive characteristics to other ingroup members.

Role-based trust: often it is not so much the person in the role that is trusted but the system of expertise that produces and maintains role-appropriate behavior. In this sense trust can be seen also as depending on the system that is represented in the role – roles lessen the need for repeatedly negotiating trust when interacting with others. A related issue here is that serious failures of cooperation can occur if novel situations break down role-based habits e.g. in organizational crisis

Rule-based trust: trust based on internalized rules rests not on an explicit contract but on socialization into the structure and practices of the organization. If socialization is high and common principles are well internalized, mutual trust can acquire a taken-for-granted quality.

2.2. Developmental stages of trust

Luhmann [6] has presented a useful distinction of the antecedent conditions for the development of trust. The development of trust can be seen as depending upon two previous stages, namely familiarity and confidence. All three stages represent qualitatively different modes of asserting expectations towards the behavior of others. Familiarity is the first necessary condition for the development of trust, because no stable expectations can be formed towards the strange, which remains mentally uncontrollable. The second condition is the stage of confidence, which depends in turn upon a certain amount of familiarity of the target. Confidence is based upon expectations of normal practices, standard operations and definite norms that are supported by sanctions. Confident expectations mean that no alternative ways of doing things are actively thought about; instead, a certain scenario is taken for granted. Finally, trust is the stage where open negotiation and active search for alternatives become possible, but only if the stages of familiarity and confidence are fulfilled and do not let the trusting parties down. [6, also 10]

3. Data and methods

The data was collected in an interview study that aimed at exploring working practices and problems in global inter-organizational software development projects. The focus of the study was on networked projects that involve at least two companies: a customer and a supplier. Nevertheless, most networks studied involved more than two organizational parties.

The data was collected from nine distributed software development networks. In each case the customer company was Finnish, and all of them, except one, were large and nationally well known. Altogether eight customers, five subcontractors and ten projects were studied. The data consists of taped and transcribed thematic interviews of the project personnel and managers (N=44).

Of the networks, four were developing software products, two developed bespoke systems, and three developed embedded systems.

4. Results and discussion

On the basis of the interviews it seems that most of the major problems in networked projects are related to communication and the arrangement of cooperation between companies. Due to problems in these areas, projects were easily delayed or even failed. In our interview study we noticed that companies were very interested in networked product and software development, but because of the fear of possible problems and the lack of concrete working procedures they were sometimes hesitant to start that kind of projects.

Section 4.1 presents the results from the analysis of the problems encountered in trust building. The classification

of problems is based on the framework of potential sources of trust presented in section 2.1.

Section 4.2 outlines the practices found in interviews that proved to be useful in tackling the problems. These successful practices are classified according to the developmental stages of trust described in section 2.2.

4.1. Problems in trust building

4.1.1. Personal dispositions. The receiver did not always know how to interpret the messages (e.g. e-mail) from senders in different companies, because the personalities and interaction styles of parties remained ambiguous. Some persons preferred short-worded mails that went straight to the point, but if the receivers had never really got acquainted with the sender, they easily thought that the sender was not satisfied with their work or did not respect their ideas. Situations got worse, if receivers interpreted the messages as including unwarranted commanding functions, when after all the case in point was that the personal ways of expressing oneself in mediated communication differed so significantly.

4.1.2. Common history. Because of the temporary nature of software development networks, the exchange of background information was often not sufficient, especially if companies had not properly planned and documented what kind of information would be required in each development phase. The companies often forgot to discuss the available documentation and whom to contact in specific issues. If clear organizational charts and face-to-face meetings were lacking, people did not get to know each other's roles, responsibilities and competences. Thus people hesitated to spontaneously give and ask for help. In some cases customers made impossible demands to subcontractors, who lacked the required background information; as the subcontractor then presented their own solutions to the tasks, the customer claimed that the subcontractor was incompetent and had to redo the whole task. These kinds of violations of reciprocity led easily to a decline in trusting attitudes and motivation.

4.1.3. Mediating third parties. In collocated projects tacit knowledge, positive experiences and reasons for successes and failures are spread automatically in various informal occasions and conversations increasing the awareness of the progress of the project and of possible sources of risk and opportunity. However, in the case of networks spontaneous transfer of knowledge via third parties was often blocked, because no mediating link persons between companies were available. Parties remained ignorant about reasons for delays in deliveries and testing. Similarly, the causes for changes and some troubling bugs remained unclear. This kind of uncertainty aroused suspicions about the positive intentions and motives of other parties, ending up in unwarranted accusations towards remote teams. The absence of mediating third parties also manifested as question overloads. When people did not know whom to ask for help, it was typically one salient key person (e.g. a system architect) who received a huge load of questions. In such cases the work of the contacted link person suffered heavily, and crucial information was left untransmitted.

4.1.4. Shared category membership. One obvious problem was to create a sense of togetherness at the network level. In many cases, people from different companies did not actually feel that they were working towards a common goal. Rather, the sub-goals of each site tended to conflict with commitment to a common goal. Uncertainty existed concerning the limits of confidential information that should be withheld from other companies. When in doubt, team members preferred to withhold all information and not to exchange ideas that would have helped the progress at the whole network level. Further, when, e.g., subcontractors did not get feedback on the quality of their work and could not perceive how their contributions affected the progress of the whole project, it became even more difficult for them to identify with a common goal. As a consequence, the commitment of the subcontractor was weakened and in a couple of cases collapsed totally. Problems were raised by the mere dissimilarity of the deliverables coming from "alien" sites: in one case testers who encountered deficiencies in code coming from another site intentionally made impossible change requests and finally refused to test the code from that particular site.

4.1.5. Predictable role behavior. No one of the studied networks had established role definitions at the level of the entire network. Companies may have had indicated roles in their internal processes, but at the network level a clear prediction of the behaviour of other people on the basis of their roles was not possible. Difficulties were also related, e.g., to the contested justification of decisions, because the roles did not suggest who had the right to decide on issues, especially at lower levels in the organization. Sometimes developers made confusing changes to the core modules of the product, even though that kind of tasks were not ascribed to them. At times information passed over crucial persons (e.g. an architect) because task dependencies were not reflected in the agreed communication relations between roles.

4.1.6. Internalized common rules. It was quite surprising that such basic issues as common terms to be used in the development process were not always clearly stated. Similarly, some parties were for a long time ignorant of the true nature of the development process and development cycle in other companies. The lack of binding principles manifested, e.g., as lacking communication and change-request protocols. There was only seldom agreed reaction times to received mails, which caused confusions

in communication. An unclear threshold for changes caused unnecessary and overlapping changes.

4.1.7. Discussion. On the basis of the presented classification of problems in trust building it seems that the traditional sources of trust cannot properly function in networked conditions. There are simply very few natural sources of trust that would facilitate the cooperation between parties. Especially commanding communicative acts (e.g. giving orders, asking for help, delegating tasks) between parties may lead to difficult situations in absence of underlying trusting attitudes. In this sense, the building of trust in global software development is something that must be intentionally arranged and taken care of.

4.2. Practices for successful trust building

The interviewed companies had managed to establish some practices that proved to be successful in building trust. The practices were rather unsystematically implemented and, accordingly, the results for trust building were not optimal.

4.2.1. Practices supporting the development of familiarity. A common kick-off meeting in the beginning of the project was a successful way to create initial familiarity between the members. Successful kick-off meetings did not have to include all the members participating in a single event; instead there could be several kick-off meetings at different times associated to, e.g., interdependent subprojects. The important thing was that those individuals who would be changing information or cooperating with each other, would get to know each other at the beginning of the project.

In some cases cooperation problems were solved only after one party (e.g. the customer) had visited the other party's (e.g. the subcontractor's) premises and got acquainted with the working process and the nature of the encountered problems. In this sense collocated reviews, training occasions and joint planning meetings were invaluable for both parties, because in these meetings great amounts of background information and tacit knowledge could be exchanged.

Other practices that facilitated the development of familiarity included the establishment and updating of an organizational chart for all the members to see, e.g., on the project web pages. A useful organizational chart included information about project members in all companies, e.g., names, roles, pictures, and contact information. Additionally, salient informing about the project goals in the beginning of the project – e.g., in the form of project plans or start-up meetings – also provided a ground for forming a common identity and vision of the project.

Together these practices function to create a basis for the first step in trust building, namely familiarity. Some of these practices may seem quite trivial, but precisely because of the lack of traditional sources of trust, a proper implementation of the aforementioned practices proved to be crucial in networked conditions.

4.2.2. Practices supporting the development of confidence. A tricky issue regarding the development of confidence in the studied cases was to establish binding and clear inter-organizational processes and stabilizing structures across the network. A useful practice in the beginning of the project was a collocated training of the development process to be used: things progressed more fluently after the exact meaning of the terms to be used in the development process was clearly agreed on between parties. Issues that were informed only in writing were often improperly internalized. Another related step was to give detailed feedback of the encountered deficiencies in code. If illustrative feedback of the deficiencies was given, perhaps face-to-face by a liaison person, the initial failures turned out to be strengths in some cases. Internalization of the coding and documentation principles was better when trained after some insufficient trials.

Another major area in need of stabilizing practices was the arrangement of inter-organizational communication. A promising practice that was being developed in a couple of cases was the allocation of, e.g., task descriptions, decision-making rights and responsibilities to specific roles. These roles could be linked to matching roles in other companies. This way, the interdependencies between tasks and the exchange of crucial information became more clearly structured, providing a more predictable environment for the development work and helping people find the correct persons in important issues, preventing effectively both information overloads and information blocks. Further, clearly agreed reaction times to received e-mails, messages and questions decreased confusion between parties.

When applied systematically, these practices created expectations of normal routines, standard operations and definite norms that guaranteed that the process kept on going solidly even in the face of problems and unexpected changes. In other words, when properly implemented, the practices established confidence, which can be seen as the second step on the way to successful trust building.

4.2.3. Practices supporting the maintenance of trust. In the cases studied, a successful practice in maintaining a trusting orientation was proper informing about the project progress to all contributing parties. Mere follow-up based on reported working hours did not suffice, since the parties wanted to know how their contributions affected the progress of the whole project. Instead, feedback about the quality and concrete contributions of the deliverables was appreciated. When the delivering parties could recognize, what had gone especially well and what were the reasons for possible dissatisfaction, their working morale

and motivation remained high. Also, the exchange of experiences in the development work across team and company borders (e.g. chat, phone, e-mail lists) helped to create a common understanding and lowered the threshold for spontaneous offers of helping acts. When, e.g., developers learned about the circumstances and difficulties on other sites, their suspicions towards the alien deliverables were lowered. However, without a supporting management policy an open atmosphere was not likely to develop.

4.2.4. Discussion. When the identified practices are viewed from the perspective of trust building, practices providing familiarity work as compensators for the first four sources of trust by providing knowledge of personal dispositions, helping to build a common history, introducing mediating third parties and identifying salient memberships in shared categories. Second, the practices that establish confidence compensate the most obvious deficiencies in the last two source categories by facilitating predictable role behaviour and the internalization of common rules. Third, if open and negotiating communication prevail and the parties are willing to inform each other about the project progress while also accepting the right to assess the contributions of each other, a basis for maintaining a trusting orientation is laid down.

5. Conclusions and future research

On the basis of both literature and the empirical findings it seems that the conditions associated with distributed global software development are very demanding, because most of the traditional sources of trust don't exist in networked conditions. Consequently, trust in networks may emerge occasionally, but maintaining it is especially challenging. However, by concentrating properly on the practices that support the development of antecedent conditions of trust – familiarity and confidence – in the beginning of projects, trust building may bring out successful results also in the case of geographically and organizationally dispersed networks.

The paper provides only initial outlines of the nature of problems and possible solutions in trust building. In the future our aim is to analyze the nature of the problems more thoroughly, e.g., by studying the typical sources of problems in different kinds of software development project types. We believe that when the challenges of cooperation in different project types are understood better, it also becomes possible to formulate adequate supporting practices for different kinds of networks and projects. Nevertheless, since it seems that the traditional sources of trust do not exist in networked conditions, it is probable that all types of global inter-organizational software development projects will benefit from some basic and general practices that support trust building.

6. References

[1] Creed, D.W.E., and R.E. Miles, "Trust in organizations", in Kramer, R.M., and T.R Tyler, (eds.) *Trust in organizations*, Thousand Oaks, Sage, 1996, pp. 16-38.

[2] Heeks, R., S. Krishna, B. Nicholsen, and S. Sahay, "Synching or sinking: Global software outsourcing relationships", *IEEE Software*, March/April, 2001.

[3] Herbsleb, J., A. Mockus, T. Finholt, and R. Grinter, "An empirical study of global software development: Distance and speed", *Proceedings of the 23rd International Conference on Software Engineering*, ICSE 2001, pp. 81-90.

[4] Jarvenpaa, S.L., K. Knoll, and D.E. Leidner, "Is Anybody Out There? Antecedents of Trust in Global Virtual Teams", *Journal of Management Information Systems*, Vol. 14, No. 4, 1998, pp. 29-64.

[5] Kramer, R.M., "Trust and distrust in organizations: Emerging perspectives, enduring questions", *Annual Review of Psychology*, Vol. 50, 1999, pp. 569-597.

[6] Luhmann, N., "Familiarity, confidence, trust: problems and alternatives", in Gambetta, D. (ed.) *Trust: Making and Breaking Cooperative Relations*, Basil Blackwell, Oxford, 1988, pp. 95-107.

[7] Miles, R.E., and C.C. Snow, "Causes of failure in network organizations", *California Management Review*, Vol. 34, No. 4, 1992, pp. 53-72.

[8] Mishra, A.K., "Organizational responses to crisis", in Kramer, R.M., and Tyler, T.R., (eds.) *Trust in organizations*, Thousand Oaks, Sage, 1996, pp. 261-287.

[9] Sabherwal, R., "The role of trust in outsourced IS development projects", *Communications of the ACM*, Vol. 42, No. 2, 1999, pp. 80-86.

[10] Seligman, A.B., "Trust and sociability: On the limits of confidence and role expectations" *American Journal of Economics & Sociology*, Vol. 57, No. 4, 1998, pp. 391-404.

[11] van der Smagt, T., "Enhancing virtual teams: social relations vs. communication technology", *Industrial Management & Data Systems*, Vol. 100, No. 4, 2000, pp. 148-156.