Designing Tools to Support Trust in Distributed Software Teams

Sabrina Marczak[1] sabrina.marczak@pucrs.br

Ban Al-Ani[2] balani@ics.uci.edu

David Redmiles[2] redmiles@ics.uci.edu

Rafael Prikladnicki[1] rafael.prikladnicki@pucrs.br

[1]Faculdade de Informática
Pontifícia Universidade Católica do Rio
Grande do Sul (PUCRS)
Porto Alegre, RS, Brazil

[2]Department of Informatics
University of California, Irvine
Irvine, CA 92697-3425, USA

ABSTRACT

Software development is an inherently collaborative activity requiring team members to coordinate and communicate with each other in order to accomplish their tasks. Software development in a distributed setting exacerbates the challenges of collaborating as team members are forced to interact with remote colleagues and emphasizes the importance of trust. We posit that tools can support the development of trust is distributed teams; however, we also recognize that the understanding of how to develop such tools is still limited. This position paper argues that we need to further the understanding of what features and software requirements are needed to better support trust development and to subsequently design better tools for such a purpose.

Author Keywords

Trust; global software development; distributed teams; collaboration; software tools; tool features; tool requirements.

ACM Classification Keywords

H5.3 Group and Organization Interface: Computer-supported cooperative work; K.4.3 Computers and Society: Organizational Impacts: Computer-supported collaborative work.

General Terms

Human Factors; Design

INTRODUCTION

Trust is an essential element of effective collaborations and is often more difficult to establish amongst distributed team members [1]. While we find that many researchers have explored the meaning, categories, and phases of trust, we also find that we have a limited understanding of how the development of trust can be supported.

Members of global teams often lack shared context, knowledge of others' work progress, among others [2]. Tool support is imperative to achieve successful collaboration and to help these team members to develop and to maintain trust in such settings. Past research has extensively investigated the role of tools in supporting virtual collaboration in global teams. However, the understanding of whether the tools support can help developing trust is still limited. For example, recent research has identified that Web 2.0 technologies are useful but underused [3] and that the visualization of collaboration traces might influence individuals' sense of trust toward others [4]. Furthermore, a large number of studies are still conducted with students as participants in controlled settings, limiting the understanding of real needs from those who deal with collaboration and trust issues in a daily basis.

This position paper argues that there is a need to further understand what features and software requirements are necessary to better support trust development. This argument is proposed based on our extensive review of literature on the topic and on our own empirical experience of distributed software teams.

TRUST IN DISTRIBUTED SOFTWARE TEAMS

Trust is a topic of interest of several disciplines, such as psychology, sociology, philosophy, and computer science. It can be examined at different levels—individual, team, institution, for example. Therefore, various definitions for trust exist. For instance, Furst et al. [5] conclude that an individual's trust in their team refers to the likelihood that team members will live up to expectations, whereas Sabherwal [6] considers trust a state involving confident positive expectations about another's motives with respect to oneself in situations entailing risk. Jarvenpaa et al. [7] understands that trust is composed of a rational and a social component. Rational trust is the perspective in which individuals engage in less self-protective actions and are more likely to take risks while the social perspective of trust is defined as a social duty. People will act a certain way to support others because they feel it is the right thing to do or because they feel it is their moral duty.

CSCW '13, February 23–27, 2013, San Antonio, Texas, USA.

From these definitions, we understand trust as the belief that the trustee will meet the positive expectations of the trustor [1]. These expectations form as a result of actions and behaviors, and can influence the level of trust a member poses in another. It is implicit that trust refers to integrity such that the trustee will not take advantage of the trustor, maintaining a healthy collaboration.

Trust is often an issue for virtual global teams because members may not have had the experience working with one another [8], therefore, lacking knowledge to define realistic expectations about others' actions. Members of such teams have to adapt to the diversity in culture, geographic, temporal, and organizational backgrounds, and physical distance. These differences compose the main obstacle for trust achievement in distributed teams.

Poor socialization and socio-cultural fit, increased monitoring, inconsistent work practices, reduction of communication, unpredictable communication, lack of face-to-face meetings, conflict handling, and poor language skills are reasons that cause lack, injury, or lost of trust in distributed software teams [9]. While trust can lead to effective and efficient collaborations, the lack of trust can lead to a decrease in productivity, quality, information exchange and feedback, morale among the employees, and cause an increase in relationship conflicts [9].

TRUST DEVELOPMENT AND TOOL SUPPORT

Tool support is imperative to achieve successful collaboration and to help these team members to develop and to maintain trust in distributed settings. Although tools to support collaboration in such settings have been extensively investigated, little is known about how tools can help distributed teams to develop and maintain trust.

Based on the assumption that informal communication is the means by which people normally discover facts about their co-workers, thus leading to their mutual trust, Aranda and colleagues [10] have designed a tool called Trusty to support the development of trust in distributed teams. Trusty provides mechanisms to support communication, coordination, informal knowledge sharing, and to generate reports and statistical analysis about the social networks established through the tool in order to help project leaders to obtain feedback about members' interactions. The tool was designed based on data collected through a survey with practitioners of 4 distinct companies located in Spain and in the US. Although 23 of the 36 people targeted in the survey responded to it, only 9 of them had experience working with distributed teams [11].

Calefato and Lanubile [12][13] developed a tool to extend a commercial collaborative development environment, the Microsoft Visual Studio and Team Foundation System, to provide social awareness support for reducing sociocultural distance. Social awareness is understood as the information that a person maintains about others in a social or conversational context. It contributes to counteract the lack of teamness and strengthen trust among remote developers. They argued that disseminating social awareness information through such type of environment can both speed up the establishment of a cross-organizational shared context and help developers who have little or no chances to meet and discuss personal matters with colleagues develop trust-based inter-personal connections. The tool, named SocialTFS, is composed of three main components: one component that handles the visualization of all the social content collected from the services enabled by a user (such as Twitter and Facebook), another component that notifies events and workspace changes, and a third component that retrieves information about registered users from social network services and about other software projects in the collaborative environment. Because the information collected from the networks and project databases are fluid and change over time depending on task assignments or development phases, the tool was designed to provide dynamic followings, which are automatically added and removed to the user's environment. The tool still needs to be empirically evaluated by large-scale industrial projects that use the Microsoft Team Foundation System in order to its usefulness be assessed.

Collaboration is often understood through the identification of interdependencies among team members (e.g., task dependency); thus some researchers assume that the visualization of dependencies can lead to an understanding of other team members' workload and their ability to meet commitments. Establishing expectations of other members can increase the likelihood of them meeting these expectations and, therefore, increasing trust in these members. To explore these assumptions, Erik and colleagues [14] conducted a controlled experiment simulating a global software project with students utilizing Ariadne [15] as a tool to generate the visualization traces. Collaborative traces are representations of past and current activity of a group of developers manipulating software development artifacts. For instance, project meta-data such as the distribution of source-code authorship, work item and module assignments are examples of collaborative traces. They found evidence that Ariadne can engender trustworthiness towards developers whose code is highly dependent upon by others. However, some participants were concerned about information bias and questioned what the visualization left out (e.g. aspect relating to an individual's reliability). Others were more concerned about avoiding bias and giving individuals an opportunity "to start fresh". Despite these concerns, they conclude that the visualization of collaborative traces can help the development of trust in distributed teams. The experiment was conducted with a small number of subjects who were provided with limited information about a fictitious global software team, reducing the reliability of the findings.

Erik and Redmiles [4] recently extended their previous work by providing design principles for software tools intended to support the development of trust in distributed teams. They propose a design space for such tools that consists of three elements: trust factors, collaborative traces, and visual representations. Trust factors are aspects of work shown in the research literature to influence one's perceived trustworthiness of their team members. Examples of trust factors are team size, project type, diversity, leadership, and expertise. Collaborative traces, as previously defined, are representations of past and current work done by team members manipulating project artifacts. Collaborative traces provide information about a trust factor. Visual representations consist of a set of visual abstractions of the collaborative traces arranged in a layout that provides users with the ability to formulate accurate perceptions of their team members' trustworthiness with respect to a particular trust factor. They apply elements from the design space to the design of three example visualizations in order to exemplify their design principles. Real-case tools need to be designed now based on such principles for the demonstration of their applicability.

Web 2.0 technologies, such as wikis, blogs, tags, podcasts, feeds and social networks, can provide support for collaboration in large and small organizational teams. Therefore, it is expected that such technologies will also support the development of trust in such teams. In a recent study [16], Al-Ani and colleagues found that the adoption of Web 2.0 technologies to support the development of trust by developers of globally distributed teams is still limited. The lack of alignment with work practices and the mistrust in the information made available through these tools are among the reasons claimed by the study participants to justify the non-use of such technologies. Some participants reported that reliance on software application to develop trust with others is minimal because they believe information might not be up-to-date or might never replace the way people establish trust in person. In a follow-up investigation [3], they found that age, experience working in distributed development, and the adoption of diverse communication tools are other factors affecting the use and non-use of Web 2.0 technologies in the development of trust in real global software teams. Replication of this study in organizations with different characteristics would allow for a more comprehensive list of factors that affect the adoption of Web 2.0 technologies in global settings.

DISCUSSION

Despite the research efforts presented in the previous section, we see that little is still known of how tools can help develop and maintain trust in distributed software teams. The perspective from practitioners is even more restrictive, those who face issues related to getting to know a colleague that is remotely located and often not accessible during working hours reducing collaboration and the difficulties of establishing trust with one another. Empirical studies conducted with students provide insights and shed some light on prospective theories. However, they lack the richness of real software development environments. The lack of such details may jeopardize how accurately findings from students' studies reflect what happens on the field.

Although some recent studies are breaking new grounds, such as Al-Ani and colleagues' investigation about the adoption of Web 2.0 technologies in the development of trust [3], limited knowledge has been acquired about why such technologies are still not broadly adopted by globally distributed developers. They concluded from their study that the majority of the participants felt the effort expended in using such applications far outweighed the benefits. The lack of trust in such technologies also come from the fact that developers believe information might not be up-to-date or that they might never replace the way people establish trust in person. These findings lead us to raise questions such as "*What can be done to turn information in tools that support trust development reliable and less effort-expensive for the users?*", "*How can we replicate the way people establish trust in person through tools*?".

Evidence that tools can be utilized to support the development of trust is concluded from the experiment conducted by Erik and Redmiles [14]. The participants of this study, graduate students, reported that visualization traces may not necessarily help them develop trust in some instances. They were concerned about information bias and about giving individuals an opportunity "to reset" their past behavior and "to start fresh". Therefore, we ask "*To what extent information recovered from software repositories can be used to support distributed team members to build trust*?", "*To what extent past information about a member might influence how this member is currently perceived by his/her colleagues and can then influence the trust building process of those who have never worked together before?*".

Another interesting aspect to be observed is the focus of the tools developed specifically to support trust development. Trusty [10] is focused in supporting communication while SocialTSF [12] focuses in fostering the sharing of social awareness. Communication and awareness are two important aspects of collaboration and are factors that affect the way people establish trust. In their recent work Erik and Redmiles [4] derived a vast list of factors that affect trust building from existing literature. We find that current tools address only a few of these trust engendering factors. These findings prompted us to ask: "*How can we account for and model other factors that influence trust development in the current tools or in future tools?*", "*Which features and software requirements are necessary to support these factors?*". A more broader question can also be asked: "*What makes a tool explicit for the support of trust building?*". These are open-ended questions that need to be further addressed in order for us to move toward a better design of tools that can help distributed team members trust each other and, as a consequence, maintaining or increasing productivity, quality, and other project performance measures expected by managers to consider a project successful.

In addition, future work claims for the replication of studies in real settings aiming to collect the needs from those who practice software development in a daily basis. We also need to expand investigations to observe cases with distinct characterizations and from different locations aiming to better understand the diversity of software development as a whole, making it possible to design tools more prone to attend the needs of an entire group of professionals.

CONCLUSION

We have identified studies that have proposed tools to support the development of trust by facilitating the way that members of distributed teams communicate, coordinate, and share social awareness. We have learned that the adoption of Web 2.0 technologies to support the development of trust is still not as broad as one would expect. In addition, our literature review also revealed that collaborative traces can support trust development. Principles for tools design were derived from such preliminary findings. Although these studies corroborate our assumption that tools can support the development of trust, little is still known about how such tools should be developed. Therefore, we raised some questions that can be used to guide future studies. These questions ask about how we can turn these tools in more reliable sources of reliable information, how tools can replicate the way people establish trust in person, to what extent information recovered from software repositories can be used to support trust development, and what is the extent of influence that such information cause in team members which, as a consequence, affects the level of trust in their colleagues.

AUTHOR BIOGRAPHIES

Sabrina Marczak is an Adjunct Professor at the Computer Science School at PUCRS, Brazil. She is mainly interested in how soft issues such as trust impact collaboration practices in distributed teams. This workshop provides her with a great opportunity to discuss the matter with peers.

Ban Al-Ani is a Research Scientist at the Donald Bren School of Information and Computer Sciences at UC Irvine. Her research focuses on understanding users of technology in different contexts. The workshop provides an opportunity to participate in discussions regarding trust with others in the community, and build on her existing knowledge.

David Redmiles is a Professor in the Department of Informatics in the Donald Bren School of Information and Computer Sciences and a member of the Institute for Software Research (ISR) at UCI. He is interested in workshop participants' perspectives on the trust factors that affect distributed development.

Rafael Prikladnicki is a Professor at the Computer Science School at PUCRS, Brazil. His main research interest lies in in distributed software development. This workshop provides him with the opportunity to learn from other participants.

REFERENCES

1. Al-Ani, B. and Redmiles, D. In strangers we trust? Findings of an empirical study of distributed development. In ICGSE, Limerick, (2009), 121-130.
2. Ehrlich K. and Chang K. Leveraging expertise in global software teams: Going outside boundaries. In Proc. IEEE ICGSE, Florianópolis, Brazil (2006), 149–158.
3. Al-Ani, B., Wang, O., Marczak, S., Trainer, E., and Redmiles, D. Distributed developers and the non-use of Web 2.0 technologies: A proclivity model. In Proc. IEEE ICGSE, Porto Alegre, Brazil (2012), 104–113.
4. Trainer, E. and Redmiles, D. Foundations for the design of visualizations that support trust in distributed teams. In Proc. ACM AVI, Capri Island, Italy (2012), 34-41.
5. Furst, S., Blackburn, R., and Rosen, B. Virtual team effectiveness: A proposed research agenda. *Information Systems Journal*, 9 (1999), 249-269.
6. Sabherwal, R. The role of trust in outsourced IS development projects. *Com. of ACM*, 42, (1999), 80-86.
7. Jarvenpaa S., Knoll, K., and Leidner, D. Is anybody out there? Antecedents of trust in global virtual teams. *J. of Management Information Systems*, 14, 4 (1998), 29-64.
8. Walther, J. and Bunz, U. The rules of virtual groups: Trust, liking, and performance in computer-mediated communication. *J. of Communic..*, 55 (2005), 828-846.
9. Moe, N. and Šmite, D. Understanding a lack of trust in global software teams: A multiple-case study. *Software Process Improvement and Practice* (2008), 217–231.
10. Aranda, G, Vizcaíno, A., Hernández, J. L., Palacio, R., and Morán, A. Trusty: a tool to improve communication and collaboration in DSD. In CRIWG, RJ, Brazil, 2011.
11. Aranda, G., Vizcaíno, A., Palacio, R., and Morán, A. What information would you like to know about your co-worker? In ICGSE, Princeton, US (2010), 135–144.
12. Calefato, F. and Lanubile, F. Augmenting social awareness in a collaborative development environment. In CHASE, Zurich, Switzerland (2012), 12-14.
13. Calefato, F., Lanubile, F., Sanitate, N., and Santoro, G. Augmenting social awareness in a collaborative development environment. In Proc SSE Workshop, New York, USA (2011), 39-42.
14. Trainer, E., Al-Ani, B., and Redmiles, D. Impact of collaborative traces on trustworthiness. In Proc CHASE, New York, USA (2011), 40-47.
15. de Souza, C., Quirk, S., Trainer, E., and Redmiles, D. Supporting collaborative software development through the visualization of socio-technical dependencies. In SGW, Sanibel Island, USA (2007), 147-156.
16. Al-Ani, B., Marczak, S., Trainer, E., Redmiles, D., and Prikladnicki, R. Distributed developers' perspectives of Web 2.0 technologies in supporting the development of trust. In The Future of CSD Workshop, Seatle (2012).