Can Social Awareness Foster Trust Building in Global Software Teams?

Fabio Calefato
University of Bari
Dipartimento di Informatica
Bari, Italy
fabio.calefato@uniba.it

Filippo Lanubile
University of Bari
Dipartimento di Informatica
Bari, Italy
filippo.lanubile@uniba.it

Francesco Sportelli
University of Bari
Dipartimento di Informatica
Bari, Italy
f.sportelli1@studenti.uniba.it

ABSTRACT
Trust is paramount in distributed software development to prevent geographically distributed sites to feel distant and act like distinct teams with own conflicting goals. Nevertheless, how to build trust among developers with few or no chances to meet is an open issue. To overcome such a challenge, we hypothesize that increased social awareness may foster trust building in global software teams.

In this paper, we present two different empirical studies, specifically designed to test our hypotheses.

Categories and Subject Descriptors
D.2.6 [**Software Engineering**]: Programming Environments – *integrated environments.*

General Terms
Human Factors.

Keywords
Social awareness; Trust; Application Lifecycle Management; ALM; Social networks; SNS.

1. INTRODUCTION
Although important to any kind of team, trust is a factor that dramatically contributes to the success or failure of large projects that run on a global scale [7]. In fact, trust is paramount in globally distributed contexts to prevent that physical distance may lead to psychological distance. Reduced trust has been reported to (a) aggravate the feeling of being separate teams with conflicting goals, (b) decrease the willingness to share information and cooperate to solve problems, and (c) affect goodwill toward others in case of objections and disagreements [1]. Trust among teams typically grows through close interaction and face-to-face (F2F) communication, since it represents the most effective way to establish connections with other group members and gain awareness of both technical aspects, such as terminology or problem-solving style, and even more subtle aspects, such as cultural diversity. Unfortunately, F2F interaction is also the very activity that global software teams see reduced. In fact, time for travelling is very limited – not every one team member can visit other sites – and budget is spent early – typically at the beginning of projects, thus not allowing enough time to establish connections, especially if projects are young [2]. Nevertheless, software development organizations have become more and more distributed over the last decade, despite the fact that no answer has been provided to the following research question*: How do we strengthen or build trust among members of globally distributed teams who have few or no chances to meet?*

Seeking for an answer, we hypothesize that the disclosure of teammates' personal interests in the context of a shared workspace may facilitate the establishment of interpersonal connections, increase the likelihood of successful interactions, and help to build trust among members of global software teams. In other words, we expect that having access to information shared on social media as well as the chance to monitor others' behavior on social networks can work as a surrogate of the social interaction occurring in informal F2F meetings, thus increasing mutual trust and reducing psychological distance.

Therefore, we have developed SocialCDE, a tool that adds social awareness to Application Lifecycle Management (ALM) platforms. In the remainder of this paper, we first present the theories, upon which we built our research model; then, we present two preliminary empirical studies designed to answer our question.

2. RELATED WORK
2.1 Trust
Trust is a complex matter to study since it involves both interpersonal relationships (e.g., cultural issues between trustee and trustor) and facets of human behavior (e.g., personal traits). To date several definitions of trust have been given. A widely used and concise definition is provided by Jarvenpaa *et al.* [8], who defined trust as the expectations of one (the trustor) that others (the trustees) will behave as expected. In other words, positive trust emerges when others' actions meet our expectation; otherwise, negative trust, or mistrust, arises. Other definitions of trust distinguish between *cognitive* (or rational) and *affective* (or social) perspectives. For example, Wilson *et al.* [14] defined

SSE'13, August 18, 2013, Saint Petersburg, Russia
Copyright 2013 ACM 978-1-4503-2313-0/13/08...$15.00
http://dx.doi.org/10.1145/2501535.2501538

cognitive trust in terms of expectations about others' competence and reliability in performing important actions that the trustor cannot monitor. Conversely, affective trust relates to reciprocal emotional ties, concerns, and care between the trustee and the trustor, which push the latter to do something for the former because it is perceived as moral duty.

Several approaches for fostering trust have been proposed, the underlying idea of which is that the process of trust building develops along several dimensions called *antecedents of trust* [8], that is, the properties of the trustee that trigger the trustor's appraisal when assessing the trustworthiness of the others. According to Jarvenpaa et al. [8], *ability* (e.g., skills, knowledge), *benevolence* (e.g., courtesy, availability), *integrity* (e.g., faithfulness, adherence to moral norms), and *predictability* (e.g., reliability, consistent behaviors) are the personal characteristics of a trustee that facilitate the establishment of the trust relationship with a trustor. More specifically, the ability and predictability dimensions are assessed by means of cognitive elaboration of personal and professional information. At the same time, affective-based appraisal leads to trust building along the dimensions of benevolence and integrity. Instead, for a trustor, it is the *propensity to trust* the trait that matters.

2.2 Awareness

The concept of awareness comes from the field of CSCW and has had a considerable influence in software engineering research as well, since it provides mechanisms to coordinate group activities [11]. Group awareness has been defined by Dourish & Bellotti, as "*an understanding of the activities of others, which provides a context for your own activity*" [5].

According to Gutwin *et al.* [6], other than on their coworkers, members of a group typically also seek information on tasks and artifacts. Following these information needs, four types of group awareness have been acknowledged so far, namely: *informal or presence awareness* (i.e., who is around and their availability), *group-structural awareness* (i.e., members' roles and teams' internal structure), *workspace awareness* (i.e., who changed a shared artifact and when), and *social awareness* (i.e., the information and the understanding that teammates have about their social connections within a group [12]).

3. RESEARCH MODEL & HYPOTHESES

Jarvenpaa & Leidner observed trust evolution in global teams interacting only through computer-mediated communication [9]. The analysis indicated that teams with low level of initial trust lacked in social communication at the beginning of projects. Conversely, teams that had high level of trust at the end of projects had an initial social focus in communication, which later diminished to make room for procedural and task-focused interactions. Consistently, we argue that disclosing personal and contextual information in the workspace can increase the feeling of similarity between distant teammates, thus fostering the *amount of social communication*, a manifestation of a higher numbers of successful interactions with *new ties* and *stronger bonds* established between members across sites. Finally, existing research by Jarvenpaa *et al.* [8] has shown that perceived individuals' *integrity*, *benevolence*, and *propensity* to trust are the relevant *antecedents of trust* in the sense that they facilitate affective trust building. Therefore, we hypothesize that (see Figure 1):

H_1 – *There is a positive relationship between the amount of social awareness gained through social media and the level of affective trust mutually established among distant teams.*

To test this hypothesis in an experiment, we need to measure the levels of the affective and cognitive trust. Since trust is a perception sensed by individuals, we have to rely on self-reported data.

Treinen & Miller-Frost [13] observed that the development of mutual trust between distant sites at the beginning of a project was paramount. In fact, they observed that, during the early stage of a project, building personal knowledge about the team and mutual trust turned out to be more important than resolving technical issues, since trust would allow to resolve future issues from afar (e.g., conference call), thus resulting in increased overall efficiency. Therefore, we hypothesize that (see Figure 1):

H_2 – *There is a positive relationship between the level of affective trust mutually established among distant teams and project performance.*

As per testing this hypothesis, we acknowledge that establishing a cause/effect relationship between trust and project performance is a challenging task, as many other confounding factors (e.g., project type, individual skills) may interfere along the process.

4. THE SOCIALCDE PROJECT

Application Lifecycle Management is a continuous process of managing the life of an application through platforms that provide a project workspace with an integrated tool set, encompassing all software development activities, such as requirements management, design, coding, testing, and release management [3]. The most popular ALM platforms, also known as Collaborative Development Environments (CDEs), support the four types of group awareness. As for the support of social awareness in ALM platforms, instead, it is either completely lacking or, when available, as in the case of Jazz and GitHub, the level provided is not as adequate as for the other forms of group awareness [10].

To address this limitation, we developed SocialCDE [4], a tool that extends Microsoft Team Foundation Server (TFS) and GitHub by disclosing information collected from most of the largest social networks available today (e.g., Twitter, LinkedIn). Our tool leverages the Social Proxy Server component, an aggregator developed to store all the information retrieved from a user's social networks accounts and software projects. As per the client side, two are the plugins available, one extending Visual Studio and the other any Eclipse-based IDE (see Figure 2).

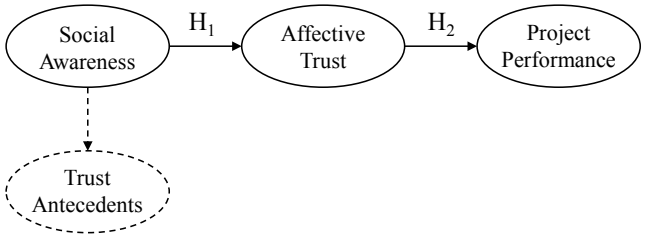

Figure 1. The proposed research model

The cached content is then requested by SocialCDE clients and presented to end users in the plugin view within the IDE. Such information is shown through three different timelines, namely *home*, *iteration*, and *interactive*. The *home timeline* resembles the same timeline available in microblogging sites such as Twitter or Yammer, as it gets populated by the posts from the current user and his/her followings. The other two timelines are the iteration and interactive timelines. The *iteration timeline* dynamically shows the content shared by any team member who reported or even commented on a work item assigned to the current user. The *interactive timeline*, instead, dynamically displays the posts from anyone who has contributed changes to the artifact opened and currently visualized in the editor view of the IDE. Finally, any new content posted stays inside SocialCDE.

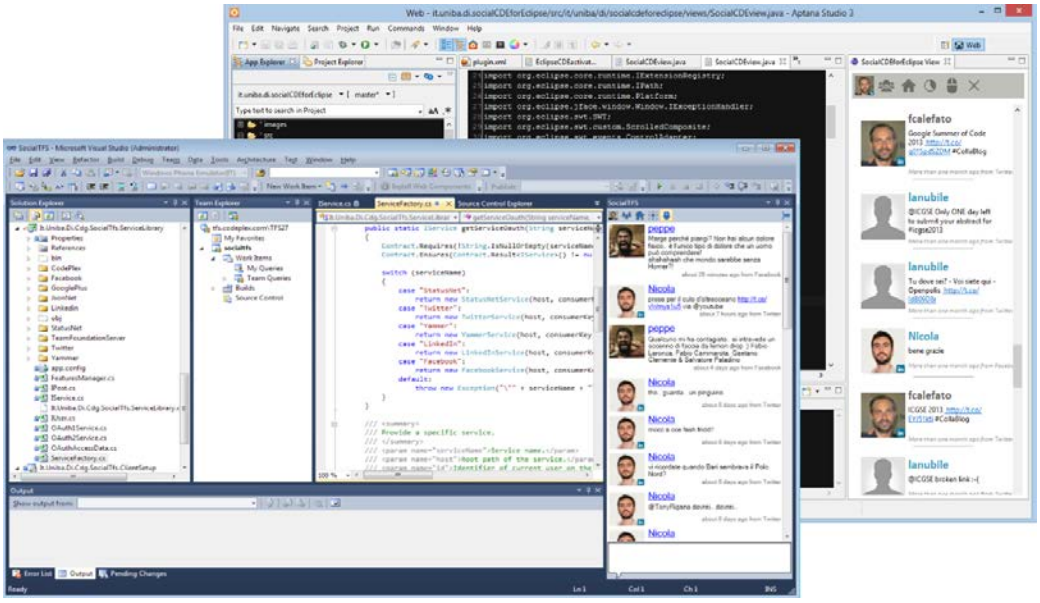

Figure 2. SocialCDE client plugins for Visual Studio (left) and Eclipse/Aptana (right)

5. STUDY DESIGNS

In this section we present two preliminary empirical studies that we have designed to start investigating our hypotheses. The first one (Study A) is a field study, involving one group, which is currently ongoing as of this writing. The second one (Study B) is a controlled experiment, that is planned to start in the summer.

5.1 Study A

This study involves a small software team of 7 members., working on "I Speak Again", a project sponsored by Informatici Senza Frontiere (ISF), a non-profit organization. We selected this project because it matches the following sociotechnical prerequisites. First, the team is distributed and not fully established. In particular, the three project leaders who know each other for several years, are able to meet face to face; instead, the remaining four members have just begun to work for the project when the study started and, besides, they are also completely distributed. Second, the project uses GitHub as ALM platform and Eclipse/Aptana as IDE. Finally, the team members agreed to connect one or more of their social network accounts to SocialCDE.

Because some of the experimental subjects involved have already been working together for some time before the experiment, we have adopted the following *A-B-A* experimental design. As shown in Figure 3, the initial stage *A* is the status the team was in before starting the experiment, when the members received instructions on how to install and use the timelines through wiki guides and videos. In addition, one of the team leaders attended a demo session in order to give support to the other developers. Before starting the experiment, all the subjects answered a preliminary questionnaire, which aimed to assess the participants' characteristics and background (e.g., age, gender, working experience, degree), as well as to measure their propensity to trust and the amount of trust, indirectly measured through the antecedents.

As of this writing, the experiment is currently in stage *B*, that is, the team has started their agile development practices and collaborative activities, using the Aptana IDE augmented with the SocialCDE plugin. This stage is going to last from four to six weeks, altough imposing strict deadlines is not feasibile because of the voluntary nature of the project. During this stage, the team members are conducting their regular work activities, plus they are using the plugin to read and share social content. Besides, usage data are being automatically and unobtrusively collected by the proxy server component (e.g., the number of posts shared). At the end of stage B, a second questionnaire will be administered to participants, with the aim of spotting differences in the perceived trustworthiness of other team members. Finally, during the third and last stage *A*, the team will go back to work with the IDE without the plugin, as in the initial stage. After one week of work without the support of our tool, one of the researchers will conduct individual, semi-structured interviews (via Skype or face-to-face), which will be recorded and transcribed. In addition, a debriefing meeting with the team leaders will also be conducted.

Given the nature of this field study, we will be able to gain data to specifically test the hypothesis H_1 on the relationship between social awareness and affective trust.

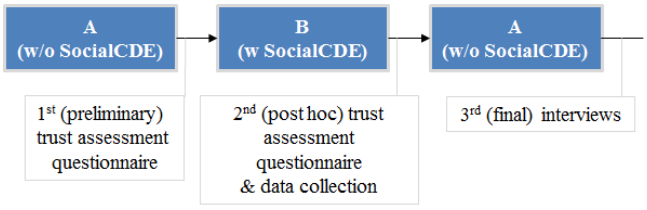

Figure 3. Design of field study A

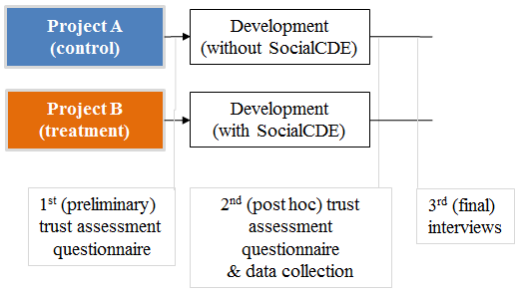

Figure 4. Design of controlled experiment B

5.2 Study B

A controlled experiment has been also designed. As compared to the previous one, this study compares two different projects with two small distributed teams of 5/6 members. Both teams will apply the SCRUM agile methodology and will develop a real software product for a customer. Team composition is the following: 3 developers (interns for a large multinational software corporation at PUCRS University in Porto Alegre, Brazil), 1 team lead (senior lecturer), and 1 or 2 product owners (customers). As in the field study, also this experiment will require the subjects involved to meet the requirement that the distributed team is newly formed. However, in this case TFS will be used as ALM platform and Visual Studio as IDE.

Since two different, but comparable, projects will be available, we will use one project (A) as the *control group* and the other project (B) as the *treatment group* (see Figure 4). More specifically, during the first stage, that is, before starting the development, the Project B team will attend an introductory lesson presenting how to use the plugin. The training lesson will be concluded with the administration of the same pre-experiment questionnaire used in the other study. Then, during the second stage, the Project A team will use Visual Studio as is, whereas the Project B team will use the IDE augmented with SocialCDE for the whole duration of the development activity (about six-eight weeks). During this development stage, the team members of both projects will have to conduct their regular work activities. Any extra activities performed by Project B team with the tool will be related to reading and sharing social content with other team members. A second post-hoc questionnaire will be administered at the end of this stage, but only to the treatment group. Finally, during the third and last stage, upon the end of the development activity for both projects, individual, semi-structured interviews will be conducted and transcribed in order to perform qualitative analysis.

For this study, we will be able to evaluate the effects of the plugin during the entire development activity of the test project and compare it to the control project. As such, it will be possible to obtain data to test both H_1 and H_2. However, as per the second hypothesis, given the small number of subjects involved, further replications will be needed to confirm any finding about differences in project performances.

6. CONCLUSIONS

In this paper, we have presented the design of two preliminary studies aimed at assessing our hypotheses that information shared on social media can surrogate the social awareness on which affective trust grows.

ACKNOWLEDGMENTS

SocialCDE is partially funded by Interreg III 2007-2013 under the project Intersocial and by Microsoft Research through the 2011 Software Engineering Innovation Foundation (SEIF) Award.

7. REFERENCES

[1] Al-Ani, B., Wilensky, H., Redmiles, D., and Simmons, E. An Understanding of the Role of Trust in Knowledge Seeking and Acceptance Practices in Distributed Development Teams, *Proc. 6th Int'l Conf. Global Soft. Eng. (ICGSE '11)*, Helsinki, Finland, Aug. 15-18, 2011, pp. 25-34.

[2] Costa, J.M.R., Cataldo, M., and de Souza, C.R.B. The Scale and Evolution of Coordination Needs in Large-Scale Distributed Projects: Implications for the Future Generation of Collaborative Tools. *Proc. Int'l Conf. on Human factors in computing systems (CHI'11)*, Vancouver, BC, Canada, May 7-12, 2011, pp. 3151-3160.

[3] Calefato F. and Lanubile F. *Practice: Collaborative Development Environments*, in Global Software and IT: A Guide to Distributed Development, Outsourcing, and Supplier Management (C. Ebert ed.), IEEE, Wiley, 2012.

[4] Calefato F. and F. Lanubile F. SocialCDE: A Social Awareness Tool for Global Software Teams. *ESEC/FSE 2013*, St. Petersburg, Russia.,18-26 Aug. 2013.

[5] Dourish P. and Bellotti, V. Awareness and Coordination in Shared Workspaces. *Proc. Int'l Conf. on Computer Supported Cooperative Work (CSCW '92)*, Toronto, Canada, Nov 1-4, 1992, pp. 107-114.

[6] Gutwin, C., Greenberg, S., and Roseman, M. Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. *Proc. HCI (HCI'96)*, London, Aug. 20-23, 1996, pp. 281-298.

[7] Herbsleb, J.D. et al. Global Software Development at Siemens: Experience from Nine Projects. *Int'l Conf. Sw Eng (ICSE'05)*, St Louis, USA, May 15-21, 2005, pp. 524-533.

[8] Jarvenpaa, S.L., Knoll, K., and Leidner, D.E. Is anybody out there? Antecedents of trust in global virtual teams. *Journal of Management Information Systems,* 14(4), pp. 29-64, 1998.

[9] Jarvenpa S.L. and Leidner D.E. Communication and Trust in Global Virtual Teams. *Journal of Organization Science*, 10(6), pp. 791-815, Nov—Dec. 1999.

[10] Lanubile, F., Calefato, F. and C. Ebert, C. Group Awareness in Global Software Eng., *IEEE Softw.*, Mar. 2013, pp.18-23.

[11] Mistrík, I., Grundy, J., van der Hoek, A., and Whitehead, J. Collaborative Software Engineering: Challenges and Prospects. *Collaborative Software Eng.*, 2010, pp. 389–403.

[12] Omoronyia, I., Ferguson, J., Roper, M., and Wood, M. A review of awareness in distributed collaborative software engineering. *Software: Practice and Experience*, 40: pp. 1107–1133, 2010.

[13] Treinen J.J. and Miller-Frost S.L. Following the sun: case studies in global software development, *IBM Syst. J.*, 45(4), Oct. 2006, pp. 773-783.

[14] Wilson, J.M.,. Strausb, S.G, and McEvily, B. All in due time: The development of trust in computer-mediated and face-to-face teams, *Organizational Behavior and Human Decision Processes*, 99(1), Jan. 2006, pp. 16–33.