

Technical Report 12-01

Distribution and Agility: It's All About Trust

Siva Dorairaj, James Noble and Petra Malik

January 11, 2012

Abstract

Trust increases team performance in software development teams. Trust among team members is imperative for the success of an Agile project. Trust is, however, affected in distributed teams. We conducted a Grounded Theory study involving 45 Agile practitioners from 28 different software companies in the USA, Western Europe, Australia and India to investigate the techniques for building trust in distributed Agile teams. We present seven eminent techniques for building trust in distributed Agile teams.

1 Introduction

Trust increases team performance in software development teams [1, 2]. Understanding the dynamics of building trust is important for developing high performance cohesive teams [3, 4, 5]. Trust among team members is imperative for the success of an Agile project [1, 6]. Trust is, however, affected in distributed teams [1, 7, 8]. Several studies point out that building trust in distributed Agile teams can be difficult [9, 10, 11]. There is little empirical studies on the techniques for building trust in distributed Agile teams [12, 13].

In this paper we investigate the following research question: “*What are the techniques for building trust in distributed Agile teams?*” We conducted a Grounded Theory study involving 45 Agile practitioners from 28 different software companies in the USA, Western Europe, Australia and India, over a period of three years. The participants discussed the impact of trust in distributed Agile teams. We discovered seven eminent techniques for building trust in distributed teams: *Create Team Identity*, *Engender Cultural Understanding*, *Keep Regular Interaction*, *Encourage Knowledge Sharing*, *Build Team Relationship*, *Promote Mutual Visiting*, and *Practice Openness and Transparency*.

The rest of the paper is structured as follows: Section 2 presents a brief background on the importance of trust for distributed Agile software development; Section 3 describes the Grounded Theory research method; Section 4 presents the results of the study; Section 5 discusses related work; Section 6 describes limitation of the study, and finally, Section 7 presents the conclusion.

2 Background

Trust has been studied in many fields such as organisational management [14], psychology [15], computer science [16], social science [17] and medical [18]. Mayer et al. [14] define trust as 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”, and characterise trust into three kinds: ability, benevolence and integrity. Ability refers to the aptitude and skills that enable an individual to be perceived as competent by team members. Benevolence refers to the extent to which an individual is believed to be willing to do good to team members beyond personal profit motive. Integrity refers to the extent to which an individual is believed to adhere to a set of principles thought to make the individual dependable and reliable. An individual is deemed trustworthy when the ability, benevolence and integrity are all perceived to be high [14].

Agile teams are collaborative and self-organizing in nature [19, 20]. The Agile Manifesto asserts that the best architectures, requirements and designs emerge from self-organizing teams [21]. The emergence of self-organizing Agile teams has increased the importance of trust in software development teams [22, 23]. Managers should trust the Agile teams to deliver business values to customers [13]. Members of the team should trust that the other members are competent, knowledgeable and collaborative to design, develop, and deliver valuable software on time and on budget [6, 22]. Conversely, trust declines when team members fail to fulfill mutual obligations, or the perception of obligations differ between some team members and the individual responsible for meeting them [24]. Lack of trust within the team affects how effectively individuals contribute to teams, reduce overall team performance, increase cycle time, create higher costs and impact product quality [25]. Several studies investigate the impact of trust in software development teams [1, 3, 26, 27, 28, 29]. There is, however, a paucity of studies into the techniques for building trust in distributed Agile software development. To address this gap, we investigate distributed Agile teams to discover the techniques for building trust.

3 Research Method

Grounded Theory (GT) is a systematic research method that emphasises the generation of theory derived from systematic and rigorous analysis of data. GT was originally developed by Barney G. Glaser and Anselm L. Strauss [30]. GT is successfully being used to study the social nature of Agile teams [8, 20, 31, 32, 33]. We chose GT as our research method for two main reasons. Firstly, GT is suitable to be used in areas that are under-explored or where a new perspective might be beneficial [34], and the literature on the impact of trust in distributed Agile teams is still scarce [12, 13]. Secondly, GT allows researchers to study social interactions and the behaviour of people [30], and Agile methods focus on people and their interactions in software development teams [35].

3.1 Data Collection

We conducted face-to-face, one-on-one interviews with our participants using open-ended questions that focused on the challenges faced by distributed Agile teams, and the strategies to overcome them. The interviews were voice recorded with the consent from the participants so that we could concentrate on the conversation. We had prepared a set of guiding questions to develop a smooth discussion with the participants. We phrased our questions cautiously so that the concerns in distributed Agile software development would emerge from the participants rather than from our own agenda. We often adjusted our questions to focus on these concerns rather than our prepared questions. In order to maintain consistency in the application of GT, all data were personally collected and analysed by the primary researcher.

3.2 Participants

We interviewed 45 Agile practitioners from 28 different software organisations in the USA, Western Europe, India and Australia. The criteria to participate in this study were: (1) four years of Agile software development experience, and (2) direct involvement in a distributed Agile project. Our participants adopted Agile methods, primarily Scrum and XP, in distributed software development. Table 1 shows the participant and project details. The project distribution varied from 2 to 4 countries, the project durations varied from 6 to 24 months, and the team size varied from 8 to 50 people on different projects. In order to get a rounded perspective, we interviewed participants from a range of different roles within the projects: Scrum Master, Agile Coach, Developer, Quality Analyst, Business Analyst, Product Owner, and Senior Management (e.g. Development Manager, Recruitment Director, and Director of Technology). Due to privacy and ethical consideration, we will only identify our participants using the codes P1 to P45.

3.3 Data Analysis

We transcribed the interviews, and used open-coding to analyse the transcripts line-by-line to explore the meaning in the data by searching for similarities and differences. Open coding breaks down, examines, compares, conceptualises and categorises the data [36]. We collated key points from the data and assigned a *code*, or a summary phrase, to each key point. Using GT's *constant comparison method* [37], we constantly compared each code with the codes from the same interview, and those from other interviews. The codes that are related to a common theme were grouped together to produce a second level of abstraction called a *concept*. As we continuously compare the codes, many fresh concepts emerge. These concepts were analysed using constant comparison to produce a third level of abstraction called a *category*. From our rigorous analysis, the concepts *Create Team Identity*, *Engender Cultural Understanding*, *Keep Regular Interaction*, *Build Team Relationship*, *Promote Mutual Visiting*, *Encourage Knowledge Sharing* and *Practice Openness and Transparency* that gave rise to the category *Building Trust*.

Table 1: Summary of Participant and Distributed Agile Project. (Agile Position: Scrum Master (SM), Agile Coach (AC), Developer (DEV), Business Analyst (BA), Quality Analyst (QA), Product Owner (PO), Senior Management (MGT))

Participant (code)	Agile Position	Project Distribution	Project Domain	Agile Methods	Team Size	Duration (months)	Sprint (weeks)
P1	DEV	USA-India	Financial Services	Scrum	8 to 10	10	2
P2	AC	USA-India	E-Commerce	Scrum & XP	12 to 14	12	2
P3	SM	USA-Western Europe-India	Mobile Application	Scrum	10	8	3
P4	AC	USA-China	Online Trading	Scrum & XP	10	8	2
P5	AC	USA-India	Media Services	Scrum & XP	8	12	2 to 3
P6	DEV	USA-UK	Hosting Services	Scrum & XP	20 to 22	8	2
P7	AC	USA-Argentina-India	Domain Services	Scrum & XP	18	6	2
P8	DEV	USA-Australia-India	Publishing	Scrum & XP	9 to 10	8	2
P9	DEV	Western Europe-Brazil	Web Search Engine	Scrum & Lean	14	24	2 to 3
P10	SM	USA-Argentina-India	Software Platform	Scrum	10 to 12	8	3
P11	SM	USA-Middle East-India	Web Services	Scrum & XP	13	10	2
P12	DEV	USA-India	Hosting Services	Scrum & XP	12	18	2
P13	SM	USA-India	Web Portal	Scrum & XP	17 to 20	5	2
P14	DEV	USA-India	E-Commerce	Scrum & XP	16 to 17	36	2
P15	QA	USA-India	E-Commerce	Scrum & XP	16	18	2
P16	SM	USA-India	E-Commerce	Scrum & XP	16	18	2
P17	DEV	USA-India	E-Commerce	Scrum & XP	16	18	2
P18	BA	UK-India	Financial Services	Scrum & XP	8	12	2
P19	DEV	USA-India	Insurance	Scrum	8 to 10	10	3
P20	MGT	Australia-India	E-Commerce	Scrum & XP	9 to 12	12	2 to 3
P21	SM	USA-Australia	Financial Services	Scrum	15	9	2
P22	SM	Australia-India	E Commerce	Scrum & XP	9 to 12	12	2 to 3
P23	QA	Japan-India	Power Distribution	Scrum	7 to 8	4	5
P24	AC	Western Europe-India	Automobile	Scrum & XP	9	5	2
P25	SM	USA-India	Information Security	Scrum & XP	24	6	3
P26	AC	USA-India	Healthcare	Scrum & XP	16	ongoing	3
P27	SM	USA-Brazil	Financial Services	Scrum & XP	30	6	2
P28	MGT	USA-India	E-Commerce	Scrum	20	18	3
P29	SM	USA-India	Social Networking	Scrum & XP	14	10	2
P30	AC	Western Europe-India	Retail	Scrum & XP	8 to 10	ongoing	2 to 3
P31	AC	UK-India	Retail	Scrum & XP	15 to 20	ongoing	3
P32	MGT	UK-South Africa	Retail	Scrum & XP	12	18	2
P33	AC	Australia-Eastern Europe-India	Recruitment	Scrum & XP	50	24	3
P34	AC	USA-India	Real Estate	Scrum & XP	6 to 8	10	2
P35	AC	USA-India	Online Payment	Scrum & XP	8	18	3
P36	QA	Canada-India	Web Services	Scrum & XP	10 to 15	18	2
P37	DEV	Western Europe-India	E-Commerce	Scrum & XP	16	4	2
P38	BA	USA-India	E-Commerce	Scrum & XP	28	ongoing	2
P39	AC	USA-India	Telecommunication	Scrum & XP	22 to 25	6 to 7	2
P40	DEV	Australia-India	Online Trading	Scrum & XP	7	6	1
P41	AC	Northern Europe-India	Retail	Scrum & XP	10 to 12	ongoing	2
P42	MGT	USA-India	Healthcare	Scrum	7	ongoing	4
P43	SM	USA-India	E-Commerce	Scrum & XP	7	ongoing	2 to 3
P44	PO	Canada-India	Cloud Computing	Scrum	10 to 12	ongoing	2 to 3
P45	MGT	USA-India	Financial Services	Scrum	10	ongoing	3

4 Results

In this section, we describe the techniques for building trust in distributed Agile teams. Figure 1 shows the concepts that gave rise to the category *Building Trust*. We present selected quotations drawn from our interviews that had led us to the emergence of the category *Building Trust*.

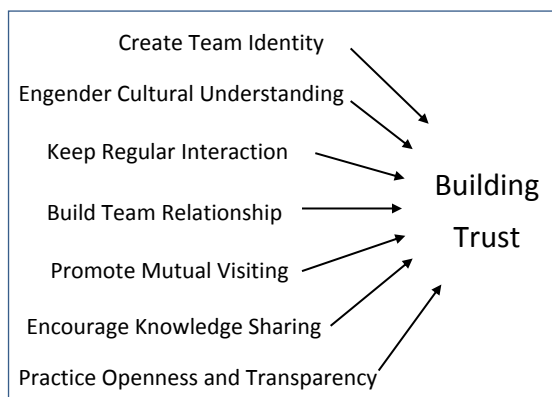


Figure 1: Emergence of category *Building Trust*

4.1 Create Team Identity

Individuals that engage in a distributed project should know the members of the teams more than just their name. Team members should have at least seen each other in photographs. Participants suggest that a sense of teamness emerge when individuals are able to identify the whole team, and the team is assured of the existence of its members. This idea of a ‘team identity’ indirectly contributes for building an initial trust in the team:

“I have not seen the entire team. We see their pictures at least. It is so important to have some kind of team identity for a team [that is] working together on a daily basis. This gets the trustability out of them working as a team.” —P24, Agile Coach.

When team members know each other, even through a photograph, they are able to initiate conversations with one another:

“Knowing the people [in the team] helps me to talk to people, [and] interact with people.” —P9, Developer.

Realising the importance of this ‘team identity’, when a new member joins an existing team or when a new team is formed, photographs of the member or the team are taken and shared with everyone in the team:

“When we form a new team itself, we take a picture of the whole team, or when a new member joins the team, the first thing we do is take a picture and share it with the other side.” —P45, Senior Management.

Participants (P1, P3, P7, P9, P16-P18, P20-P29, P32-P33, P35-P38, P44-P45) consider that creating a ‘team identity’ or a sense of teamness can foster team bonding and promote team interaction that, in turn, builds trust among the members in the teams.

4.2 Engender Cultural Understanding

The cultural diversity in distributed teams could create misunderstandings and unnecessary conflicts among team members. Individuals from different cultures face difficulties to understand team member’s interaction:

“The head wobble can often mean “I don’t know”, but it can mean “Yes!” and it can mean “No!”. I am not that good at reading exactly what all those movements are because body language is highly tied to culture.” —P22, Scrum Master.

Individuals should not adhere by a particular cultural practice that is not appropriate for the whole team. The Indian team members should not promise what they cannot deliver. Other members consider that nothing is more damaging to trust in team relationships than raising expectations which cannot be met:

“The Indians don’t say “No” to anything. This is because of the culture [that] you should always obey the seniors, or somebody with a title [or rank]. They’ll commit for something, and often they’ll not be able to meet that commitment. We need to teach them about [this] culture.” —P33, Agile Coach.

Agile practitioners ideally should attempt to understand the culture of other individuals, and not take them at face value:

“I know this [person] who would say, “Yes, I’ll have it done in 30 minutes!” when what he actually mean is that he would get it done as soon as possible. You’ve got to have the willingness to understand the [person] before you take the person at face value.” —P2, Agile Coach.

Understanding the culture of individuals in the team helps members to collaborate without unnecessary conflicts and interact meaningfully. The cultural understanding is often gained through experience, but participants suggests that training by cultural experts or experienced team members can be useful.

4.3 Keep Regular Interaction

Participant asserts that all that is needed to build the initial trust is conversation between the team members:

“Trust has to be built. We just have to have a conversation and get to know each other.”
—P21, Scrum Master.

Team members should keep a regular and continual interaction on a professional and social basis when opportunities arise, or deliberately create an opportunity to interact with team members:

“Keep talking and interacting, whether it’s in meetings, whether it’s talking to people one on one, or when pairing-programming. Social interaction is so important, and that goes a long way in building trust with the team.” —P20, Senior Management.

Daily (stand-up) meetings allow the team to meet on daily basis to share the impediments and the progress of the work. Participants benefit from daily meetings even though the meetings are short and often challenging to organise them because the teams are distributed across different time zones and, in some cases, the communication tools are limited:

“We benefited from the practice of daily meetings. When [teams] are distributed, you are definitely better of doing [daily meetings] than not doing it. We want everyone to be able to talk directly. It’s a matter of establishing trust.” —P2, Agile Coach.

Teams emphasize on verbal communication rather than written communication (e.g email or wiki) primarily to obtain a quick feedback from one another, but verbal communication also alters the nature of interaction and assists in developing interpersonal relationships. Video conferencing during stand-up meetings promotes effective verbal communication, and increases the non-verbal communication such as body language, hand gestures or facial expressions.

“Our emphasis is on verbal communication. If we have a complex question, we just pick up the phone and talk to them. We use video conferencing for stand-up meetings so that we can see the rest of the team. We make sure everyone knows each other and everyone speaks to each other regularly. It’s all for building trust in the team. ” —P27, Scrum Master.

Participants describe that computer-mediated interactions are often work-oriented and less personal. When team members were able to interact on a personal basis, the personal experience among the team members slowly built trust in the team:

“...the first 15 minutes [of daily meetings] was open time and you could talk about anything you want. And, that [was] a really good model for trust to build.” —P1, Developer.

Participants (P1-P5, P11, P19-P21, P24, P27-P30, P33, P36-P38, P40-P41) particularly acknowledged that regular interaction improves trust in the teams. The level of trust evolves over time based on a series of experiences and interactions. Regular interaction, especially using video conferencing, develops social bonding and trusted relationships among team members.

4.4 Encourage Knowledge Sharing

Some teams strongly encourage team members to participate in knowledge sharing activities such as giving presentations, discussions in forums, writing technical blogs, or proactively sharing the “*tips and tricks*” of doing a particular work. These activities provide avenues for team members to learn from one another and acknowledge that a team member is knowledgeable about a particular subject area. Knowledge sharing sessions often build ability-based trust in the team:

“We have this knowledge sharing session which happens twice in a month [where] people come up with technical or non-technical topics and give presentations. It is important, especially for developers, to set-up blogs, write articles, and participate in forums to gain trust in the team.” —P29, Scrum Master.

Based on a team spirit, team members should not withhold knowledge that are worth sharing with the team. Team members build benevolence-based trust on the members who are willing to share the project-critical knowledge that are required to carry out the project:

“The distributed teams located offshore often don’t have complete information about the project. Someone [located] onsite with the customer often contribute and coordinate to share the functional and technical knowledge across to the team. So basically this builds trust within the team.” —P30, Agile Coach.

Upon completion of a project, teams analyse the project to understand what was and wasn’t done well during the course of the project. This valuable information is useful for the teams and the organisation as a whole, and can be used for future projects to gain competitive advantage:

“Knowledge sharing happens after every project. The project team members assess project requirement and discuss through issues in the project. For a new project, this information becomes really useful. So that kind of board knowledge sharing is happening.” —P36, Quality Analyst.

As a team member shares knowledge and information with another member, the perceived trust increases between them. This implies that trust develops as a result of knowledge sharing. Participants were able to build ability-based, benevolence-based or integrity-based trust at different levels of knowledge sharing. The effective knowledge sharing mechanism within the team can foster efficient trust building in the team.

4.5 Build Team Relationship

Team members should develop strong working relationships with the individuals whom they are working with. The opportunities to build the ‘friendship’ arise when team members initially meet during project kick-off workshops. This team bonding is considered crucial for distributed teams:

“Make friends with people that you’re working with because this [relationship] helps people to trust you. There has to be a good professional working relationship. If [team] don’t trust you, distribution doesn’t work.” —P11, Scrum Master.

Team relationships is particularly important between project team and customers. Participants suggest that using video conference for daily meetings can significantly contribute for trust building between them:

“Build a working relationship with the customer, and ensure that the entire team is getting fully engaged with the customer, not just the Business Analyst or the Project Manager. To begin with, organise stand-up meetings with video conferencing. That builds a tremendous amount of trust with the customer.” —P13, Developer.

Customers should consider the project team members at all the different locations to be part of one single team. Team members should not be treated as mere workers for the customers:

“The customers should see us as their partner or their team members, and not like “They are doing some work for me because I’m paying for it.” Only then there’ll be more trust. ” —P30, Agile Coach.

Strong bonding among project team members develops good working relationships among them. Mutual respect and understanding between customers and project team members could develop long term trusted relationships.

4.6 Promote Mutual Visiting

Participants acknowledge that face-to-face interaction at the initial stage of the project is important for establishing the initial trust between the distributed members of the team:

“Face-to-face conversation, at least in the initial stage of the [distributed] project, is really important to build trust.” —P11, Scrum Master.

Though the initial trust between team members has been built, team should work to strengthen that trust for ongoing team relationships. Teams continue to foster the initial trust through mutual visits of team members in different locations on a regular basis. Participants describe that trust was high among team members that met face-to-face on a regular basis compared to those that met less often or did not meet at all:

“Agile is so built on the trust mechanism. The key thing in distributed Agile is to get people meet face-to-face at the start of the project to build that trust. And, trust has to be built continuously [because] trust breaks down really quickly. So people were travelling back and forth between different locations.” —P21, Scrum Master.

Collocation that permits team members to work in close proximity and allow frequent face-to-face interaction is imperative for the success of the distributed project:

“...get them all together [and] have them work together collocated, does not have to be for a long period of time, but people need to meet face-to-face. Otherwise, this [distribution] will not work.” —P3, Scrum Master.

Realising the importance of face-to-face interaction and working together in close proximity, managers ensure that whenever a new team is formed, one side of the team travels to the other location and works collocated for a short duration before distributing the teams:

“Whenever a new team is formed we make sure that [one side of] the team travels. They sit together and work for a while and come back. ” —P45, Senior Management.

When the team works collocated for the first few weeks of the project, or for the first iteration, the team members are able to build a strong social relationships among themselves. This initial collocation facilitates social bonding and trust building that are required when the team are distributed:

“I would collocate a team for the first few weeks of the [distributed] project before actually separating them out. This is crucial to build trust, to build [team] relationships, to build a shared understanding. It’s much easier to have conversations with people on the phone if you’ve met them previously in person.” —P22, Scrum Master.

The Indian team prefers the team members from the onsite locations to visit them rather than the other way around. During this visit to India, the team members become aware of the condition and constraints of working from the offshore locations, such as heavy traffic jams, frequent power cuts, or limited communication infrastructure. This awareness develops an understanding between team members, and could remove the barriers for building trust:

“When people from the other site come [to India] and work with us for a short period of time, they get to understand our condition and constraints, and they knew why we were asking them certain things.” —P17, Developer.

Face-to-face conversation afford participants opportunities to develop trust among each other in ways that are not possible through computer-mediated communication. Despite the inconveniences of distribution, often across different continents, teams practice mutual visiting on a regular basis throughout the project (P3, P4, P5, P21, P32, P33), or at least once during the course of the project (P1, P2, P7, P11, P22, P24, P27, P36-P38). Though participants (P6, P9, P17, P25, P29, P43-P45) did not get to travel, they consider that mutual visits are absolutely necessary for developing strong team relationships, and building trust in the team.

4.7 Practice Openness and Transparency

Participants emphasize the importance of openness and transparency in teams. Daily Scrum provides opportunities for individuals to have open and honest interaction with the team:

“During daily stand-up meetings, you’re interacting with the entire team. Everybody knows what you have done, and everybody can take a look at the code that you have written. Here, everything is open, and basically it helps in building trust. Customers trust us that we are always doing the right things.” —P14, Developer.

A team member genuinely demonstrating honesty, transparency and openness could gain integrity-based trust from the team:

“So just be honest, be transparent, [and] ensure that others know all your constraints. Openness gets people in touch with each other, [and] allows development team to come back with questions and have open communication.” —P11, Scrum Master.

Open interaction with team when conflicts arise, and respecting a collective team decision naturally allows trust to flourish in the team:

“If I don’t believe in a particular approach, I’ll have a conversation with [the team] to prove what makes sense to me, and I’ll give room for them to explain too. At the end, we always go ahead with it [based on] a ‘team decision’.” —P17, Developer.

Managers should create a supportive environment for team members to perform their jobs efficiently, and encourage them to express their opinions without fear of repercussions:

“We can reduce the fear by showing them what happens when a mistake is made. Make them aware that we are not going to blame them [but] we are actually going to solve the problem. Just be open about everything. Openness solves a lot of problems, removes blames, [and] brings responsibilities and trust in the team.” —P32, Senior Management.

Participant (P11, P17, P24, P29, P31-P34, P38) discussed that practising openness and transparency in the team can foster trusted relationship in the team.

5 Related Work

Agile methods advocates face-to-face communication and close collaboration practices including barrier-free collocated teams [38]. Scrum and XP are considered to be the most widely adopted Agile methods in software development projects [39]. Scrum advocates frequent communication through its four process activities, known as *Scrum ceremonies* [35], to keep the software development work on track. The four Scrum ceremonies are: *Sprint planning*, *Daily Scrum*, *Sprint reviews*, and *Sprint retrospectives* [35]. McHugh et al. [22] suggest that Scrum ceremonies increase trust in the team by providing transparency and visibility of project status, enhancing accountability and collective responsibilities, increasing open and frequent communication, and sharing of knowledge and obtaining feedback.

Moe and Smith [1] argue that lack of trust affects team performance and decreases team velocity. Korkala et al. [40] found that the lack of trust was one of the reasons for the customer not involved in the implementation of distributed Agile projects. Robinson and Sharp [41] argue that the sense of respect, responsibility, concern for the quality of working life and faith in ability of the Agile team would be affected without trust, and members of the team would doubt that the team as a whole could deliver valuable software to the customer.

Trust among team members is imperative for blending agility with distributed projects [8]. Substantial effort are required to build trust between the team members who are distributed in different countries, and often across different time zones. Regular face-to-face meetings and effective communication are important for building trust [11, 42]. Distributed Agile teams make conscious efforts to improve team interaction and build strong working relationships amongst the team members to allow team trust to build naturally [9]. Team members appreciate the culture training initiative that convinced the customers to trust the team for a long term working relationship [3, 43]. Frequent visits by distributed partners and sponsor visits, and building a cohesive team culture have been contributing significantly for building trust in distributed teams [8].

6 Limitation

The inherent limitation of a Grounded Theory study is that the findings are grounded in the specific contexts explored in the research [44, 45]. These contexts were dictated by the availability of the Agile practitioners to participate in this study, and by our choice of research destinations.

As we employ GT in empirical software engineering research, we often encounter criticism regarding the generalisability of the findings. We do not claim that our findings are universally generalisable to all distributed Agile software development projects, but rather our findings accurately characterize the contexts studied. In GT, claims for universal generalisability is based on the fit and transferability of the results [46]. This requires the researcher to provide sufficient description of the context in which the results were obtained, so that another researcher could employ the GT's constant comparison method on new data to determine whether a substantive results fits the new data, and the results are transferable to another situation [44, 46].

7 Conclusion

In this paper, we investigate the techniques for building trust in distributed Agile teams through a Grounded Theory study that involved 45 Agile practitioners from 28 different software companies in the USA, Western Europe, Australia and India. Building trust among distributed team members in project-oriented contexts can be difficult because individuals involved in a distributed project often have little or no prior experiences working together. This in-depth study reports on the eminent techniques for building trust though there can be other subtle trust building techniques that can be useful and effective in their own contexts, but did not emerge from our analysis. A distributed Agile team that has built trust steadily and systematically over time could meet and exceed customer expectations consistently, address impediments and problems reliably, quickly and efficiently, and establish a long term working relationship with customers and team members across different locations.

The understanding of trust in distributed Agile projects gained from this study could be used as a foundation for conducting future studies involving teams working on different Agile projects in different contexts. Further studies could also compare successful distributed Agile projects with unsuccessful projects to consider the ramifications of trust.

7.0.1 Acknowledgments.

Thanks to the Agile practitioners who participated in this study. Thanks to Dr.Rashina Hoda and Dr.George Allan for their help. This study is supported by Universiti Tenaga Nasional (Malaysia) PhD scholarship.

References

- [1] Moe, N., Šmite, D.: Understanding lacking trust in global software teams: A multi-case study. In Münch, J., Abrahamsson, P., eds.: *Product-Focused Software Process Improvement*. Volume 4589 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2007) 20–34
- [2] Davies, R., Sedley, L.: *Agile Coaching*. 1 edn. Pragmatic Bookshelf, USA (2009)
- [3] Babar, M.A., Verner, J.M., Nguyen, P.T.: Establishing and maintaining trust in software outsourcing relationships: An empirical investigation. *Journal of Systems and Software* **80**(9) (2007) 1438–1449

- [4] Lencioni, P.: *The Five Dysfunctions of a Team*. 1 edn. Jossey-Bass, San Francisco, USA (2002)
- [5] Wilson, J.M., Straus, S.G., 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) (2006) 16–33
- [6] Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to Agile methodologies. *Communications of ACM* **48** (May 2005) 72–78
- [7] Carmel, E.: Thirteen assertions for globally dispersed software development research. In: *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*. Volume 3. (1997) 445–452
- [8] Ramesh, B., Cao, L., Mohan, K., Xu, P.: Can distributed software development be Agile? *Communication of the ACM* **49**(10) (2006) 41–46
- [9] Young, C., Terashima, H.: How did we adapt Agile processes to our distributed development? In: *Proceedings of the AGILE, Los Alamitos, CA, USA, IEEE Computer Society* (2008) 304–309
- [10] Holmström, H., Fitzgerald, B., Agerfalk, P.J., Conchuir, E.O.: Agile practices reduce distance in global software development. *Information Systems Management* **23**(3) (2006) 7–18
- [11] Vax, M., Michaud, S.: Distributed Agile: Growing a practice together. In: *Proceedings of the Agile, Los Alamitos, CA, USA, IEEE Computer Society* (2008) 310–314
- [12] Hasnain, E., Hall, T.: Investigating the role of trust in Agile methods using a light weight systematic literature review. In Abrahamsson, P., Baskerville, R., Conboy, K., Fitzgerald, B., Morgan, L., Wang, X., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C., eds.: *Agile Processes in Software Engineering and Extreme Programming*. Volume 9 of *Lecture Notes in Business Information Processing*. Springer Berlin Heidelberg (2008) 204–207
- [13] McHugh, O., Conboy, K., Lang, M.: Using Agile practices to build trust in an Agile team: A case study. In Pokorny, J., Repa, V., Richta, K., Wojtkowski, W., Linger, H., Barry, C., Lang, M., eds.: *Information Systems Development*. Springer New York (2011) 503–516 10.1007/978-1-4419-9790-6_40.
- [14] Mayer, R.C., Davis, J.H., Schoorman, F.D.: An integrative model of organizational trust. *The Academy of Management Review* **20**(3) (1995) pp. 709–734
- [15] Rempel, J.K., Holmes, J.G., Zanna, M.P.: Trust in close relationships. *Journal of Personality and Social Psychology* **49** (1985) 95–112
- [16] Marsh, S.P.: *Formalising Trust as a Computational Concept*, PhD Thesis. Department of Computing Science and Mathematics, University of Stirling, Scotland, UK (1994)
- [17] Gambetta, D.: *Trust: Making and Breaking Cooperative Relations*. Blackwell Publishers (1990)

- [18] Hall, M.A.: Researching medical trust in the United States. *Journal of Health Organization and Management* **20** (2006) 456–467
- [19] Sharp, H., Robinson, H.: Collaboration and coordination in mature eXtreme programming teams. *International Journal of Human-Computer Studies* **66**(7) (2008) 506–518 Collaborative and Social Aspects of Software Development.
- [20] Hoda, R., Noble, J., Marshall, S.: Organizing self-organizing teams. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, New York, USA (2010) 285–294
- [21] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for Agile Software Development. <http://www.agilemanifesto.org/principles.html>, (last accessed on January 11, 2012)
- [22] McHugh, O., Conboy, K., Lang, M.: The impact of Agile practices on trust in software project teams. *Software, IEEE* **PP**(99) (2011) 1
- [23] Hoda, R., Noble, J., Marshall, S.: Balancing acts: Walking the Agile tightrope. In: Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering, New York, NY, USA, ACM (2010) 5–12
- [24] Piccoli, G., Ives, B.: Trust and the unintended effects of behavior control in virtual teams. *MIS Quarterly* **27**(3) (2003) 365–395
- [25] Bandow, D.: Time to create sound teamwork. *Journal of Quality and Participation* **24**(2) (2001) 41–47
- [26] Oza, N.V., Hall, T., Rainer, A., Grey, S.: Trust in software outsourcing relationships: An empirical investigation of Indian software companies. *Information and Software Technology* **48**(5) (2006) 345–354
- [27] Lander, M.C., Purvis, R.L., McCray, G.E., Leigh, W.: Trust-building mechanisms utilized in outsourced IS development projects: A case study. *Information & Management* **41**(4) (2004) 509–528
- [28] Sabherwal, R.: The role of trust in outsourced IS development projects. *Communications of ACM* **42** (February 1999) 80–86
- [29] Jarvenpaa, S.L., Shaw, T.R., Staples, D.: Toward contextualized theories of trust: The role of trust in global virtual teams. *Information Systems Research* **15** (2004)
- [30] Glaser, B.G., Strauss, A.L.: *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Sociology Press, Aldine, Chicago (1967)
- [31] Martin, A., Biddle, R., Noble, J.: The XP customer team: A grounded theory. In: Proceedings of the AGILE. (2009) 57–64

- [32] Whitworth, E., Biddle, R.: The social nature of Agile teams. In: Proceedings of the AGILE, Washington, DC, USA, IEEE Computer Society (2007) 26–36
- [33] Fox, D., Sillito, J., Maurer, F.: Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In: Agile 2008 Conference. (2008) 63–72
- [34] Schreiber, R.S., Stern, P.N.: Using Grounded Theory in Nursing. Springer Publishing, Broadway, New York (2001)
- [35] Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall PTR, Upper Saddle River, NJ, USA (2001)
- [36] Strauss, A., Corbin, J.: Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage Publications (1998)
- [37] Glaser, B.G.: The constant comparative method of qualitative analysis. *Social Problems* **12**(4) (1965) 436–445
- [38] Cockburn, A.: Agile Software Development. Addison-Wesley, Indianapolis (2002)
- [39] Fitzgerald, B., Hartnett, G., Conboy, K.: Customising Agile methods to software practices at Intel Shannon. *European Journal of Information System* **15**(2) (2006) 200–213
- [40] Korkala, M., Pikkariainen, M., Conboy, K.: Distributed Agile development: A case study of customer communication challenges. In: Agile Processes in Software Engineering and Extreme Programming. Volume 31 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2009) 161–167
- [41] Robinson, H., Sharp, H.: The characteristics of XP teams. In Eckstein, J., Baumeister, H., eds.: Extreme Programming and Agile Processes in Software Engineering. Volume 3092 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2004) 139–147
- [42] Dorairaj, S., Noble, J., Malik, P.: Effective communication in distributed agile software development teams. In Sillitti, A., Hazzan, O., Bache, E., Albaladejo, X., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C., eds.: Agile Processes in Software Engineering and Extreme Programming. Volume 77 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2011) 102–116
- [43] Dorairaj, S., Noble, J., Malik, P.: Bridging cultural differences: A grounded theory perspective. In: Proceedings of the 4th India Software Engineering Conference. ISEC '11, New York, NY, USA, ACM (2011) 3–10
- [44] Adolph, S., Hall, W., Kruchten, P.: A methodological leg to stand on: Lessons learned using grounded theory to study software development. In: Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research, New York, NY, USA, ACM (2008) 166–178

- [45] Hoda, R., Noble, J., Marshall, S.: Developing a grounded theory to explain the practices of self-organizing Agile teams. *Empirical Software Engineering* 1–31
- [46] Gasson, S.: Rigor in grounded theory research: An interpretive perspective on generating theory from qualitative field studies. In Whitman, M., Woszczyński, A., eds.: *Handbook for Information Systems Research*. Idea Group Publishing, Hershey, PA (2003) 79–102