Building trust in distributed teams by applying particular Agile practices

Jakub Skalbania[1], Angela Martin[2],

[1] Kellogg College, University of Oxford, Oxford, United Kingdom
[2] School of Computing and Mathematical Sciences, University of Waikato, Waikato, New Zealand

Abstract. Trust between team members is considered one of the most important factors to the teams' success in general. Nothing is different in IT projects. Their success frequently depends on the team cohesion and on the ways the team members can effectively co-operate. Trust between team members is particularly important between team members in distributed teams where regular, face-to-face communication and close collaboration are hard to accomplish. We used an interpretative case study to explore a successful Agile project with teams dispersed over two locations. We found that interviewees build consistent picture of how particular Agile practices can help to build trust between them as well as which Agile practices are believed to impact the level of trust between team members.

1 Introduction

"Trust needs touch." - people are used to that statement. But, what if there is no touch possible? What if the group of people is geographically dispersed over multiple locations with no face-to-face communication at all?

A number of research papers and available materials indicate that it is much easier to build trust in co-located teams than in dispersed ones [1][12], but also that building trust in distributed teams is not impossible [6][9].

Our research explores how the practices from different Agile Methods can be utilized to build trust between team members dispersed over several locations and thus to create a highly productive environment in teams that are not co-located. We have used a case study to explore an Agile project implemented by teams located on two continents. We obtained several perspectives on how the trust is perceived in distributed teams, but we also have learned how the Agile practices can be amended to increase the level of trust and the team cohesion in distributed teams.

2 Research Hypotheses

Our research hypotheses and questions were designed to help us understand whether and, if so, to which extent can Agile practices be used to build trust in distributed teams.

2.1 How Agile practices help to build trust in distributed teams?

To describe how Agile practices can be applied to distributed teams to establish trust, we must first understand what are the specific features of Agile methods that distinguish Agile methods from standard methodologies and why those features are believed to help building trust. Only then may we focus on how the particular Agile practices may help to build trust in distributed environments [3].

2.2 Which Agile practices can help to build trust the most?

Due to the nature of Agile practices, some of them simply cannot be applied to distributed environment (e.g. a need for osmotic communication) [2]. Some of them are applicable, but need to be amended. And some of them could be applied to dispersed teams 'as they are'. From those that can be utilized in distributed projects we may distinguish those that have a potential impact on trust and those that did not show any impact. It was hard to find a single practice that could be said to increase trust the most, but we may definitely say that some Agile practices (either adjusted to distributed environments or not) have a lot of positive impact on trust and team cohesion when applied to dispersed teams. This impact is briefly discussed in the following sections of this research paper.

2.3 How can particular Agile practices be adjusted for distributed teams?

We assumed the Agile practices might need to be given a re-interpretation in some cases or they may need to be slightly amended to be applicable in distributed teams. We were interested what are these adjustments and whether they are crucial for the Agile practices to build trust in distributed teams.

2.4 Research Method

We have decided to take advantage of the case study research method in order to provide a holistic view of one particular case [4] which was a distributed software development project managed with Agile methods. An interpretative in-depth case study was used to explore the research hypotheses within the setting of one software development project [7] with team distributed over two locations on two continents. The phenomenon of interest would be hard to study outside its natural setting,

therefore the interviews and observations were conducted on site, in the participants' natural environment.

Data about the case was collected through semi-structured interviews and direct observation of project participants. The participants included the tester, developer, product owner and 'business-analyst-alike' roles. The interviews consisted of up to 20 open questions, and the flow of the interviews and the depth of the questions were dependent on particular responses. The interviews were not taped, as the consent for recording them was not given by the company. The participants were given a possibility to validate the findings. This paper contains quotes from the interviews.

More information on the research method used and on more detailed study results can be found in the thesis [11].

3 Description of the Context of the Project

The aim of the project used in the case study was to develop an airline fare management system. The project involved building multiple software system components that constituted a framework for fare data exchange for multiple end customers. There was no single end customer for whom the system was built. The customers come from airline industry, and the interest in the system was shown by several companies that may benefit from it.

The approach to the system development was agile, however no specific Agile method was used. Rather, particular Agile practices were used. Not all project team members had previous exposure to any of the Agile methods, and for some of them it was the first contact with Agile.

The project was scheduled for nine months, and the development was time-boxed into two-weeks iterations. Each of them ended with a demonstration to the customer representatives (from different companies).

The project team included the team members responsible for the system development, but also the 'Customer Advocates' – subject matter experts who knew the domain and who acted as the Customers for the team.

The team was dispersed over two locations: USA and Poland. The division was not based on specific roles, so both team parts consisted of developers, testers and business analysts. The whole team consisted of 19 members: 13 people in Poland and 6 in the USA.

4 Results of the Experience

4.1 How Agile practices help to build trust in distributed teams?

The findings of the study helped to understand how Agile practices may help to build/increase trust in distributed teams. The most frequent reasons why Agile practices were believed to increase trust can be grouped as follows:

- Agile practices introduce high visibility into the project activities for all team members, thus creating an open environment where there is no need to hide from others;
- Agile practices create an environment where open, honest and informal communication between team members is possible:

"One thing, however, boosted our trust significantly – when a person, who we all knew could be responsible for the problem, openly said 'In this iteration I could have done this and that in a better way'. No one took such expressions as weaknesses. But I think everyone then felt the same – 'I could trust this person. He or she was honest even when it was not easy for him or her. I respect that'
– Developer

- Agile practices enable the team members to co-operate in an effective, informal way that best suits their current needs:

"We try to avoid planning meetings as much as possible. If I have to discuss anything, I just go to another developer and we talk to each other in a way we usually talk during our coffee breaks. No formal requirements – just pure discussion."
– Developer

- Agile practices encourage creating an environment full of technical excellence:

"Technical excellence pays off. I can't even imagine what would happen if someone without it had access to our version control system."
– Developer

4.2 Which Agile practices can help to build the most trust?

The study results show that not only soft, people-related practices help to build trust. The technical Agile practices can be as useful in increasing trust between team members as the soft ones. The answer to this research hypothesis lists the practices that emerged from the study as the ones that have positive impact on trust. The practices are described together with the adjustments made to them, so that they could be applied to the distributed project environment.

Team self-organization. Although team self-organization in geographically dispersed teams is accomplished in a different way than in co-located teams, it still seems to be crucial for the team to become and stay effective. It applies to the team members within one location, as well as to the team organization between two locations. The latter includes organizing ways to provide feedback and to organize the co-operation, so that the team members can benefit from informal communication.

The study participants noted that if authoritative management had been used in the project, the teams in both locations would have been definitely divided into 'us' and 'them'. However, because the team members were let to organize their work in the way they felt useful and they were allowed to change the team organization if needed, they felt they created an environment when others can be trusted.

> *"Because no one controls us, we have to trust each other. They all depend on me, and I depend on them. I always remember that I am not alone – we are the team."*
> – Developer

The fact that the team members were allowed to self-organize helped them to introduce some innovative and effective adjustments to Agile practices described later in this section.

Daily stand-up meetings. The daily stand-up meetings were considered by all study participants one of the most important factors in building trust in a distributed environment. The practice was significantly amended in the project under study. The team members, with the consent given by the project owners / leaders, introduced another stand-up, only for local teams. So, each day the teams took part in two very short meetings – one for every team member from both locations conducted on the phone and another one for team members within the same location. This second 'local' stand-up was confirmed to provide much bigger value to the participants, as it allowed for close face-to-face communication. However, the 'global' stand-up was believed to have increased trust, because it helped to assess the overall status of the project. The participants indicated that when they could see the team members from another location were really fulfilling their promises by doing what they planned and discussed during the stand-up, it really helped them to trust their colleagues from oversees.

The idea of the second 'local' stand-up was proposed by team members after they realized that the communication overhead they faced during the global stand-ups jeopardized their local information sharing. They complained that sometimes the communication problems took the whole time available for the stand-up and they finally started to consider such stand-ups not worth participating in them. Therefore they decided to introduce their local meeting with face-to-face open communication that was not prone to issues common for distant communication. After some time they realized that the global stand-up may be used to convey information relevant to team members from both locations and may also serve as a kind of informal status meeting for the whole project team. The developer described it in his own words:

> *"If we only had one daily stand-up, the one for the whole team, I don't think everything would work so well. I can even imagine, one common stand-up would not bring any value to us. However, because we had another local stand-up, we were much better prepared for the global daily meeting. Thus we were able to concentrate on what the other part of the team wanted to convey, instead of getting frustrated because of wasting our time on the conf-call."*
> – Developer

Communication. There were two main types of communication in the project under study – the local communication and the distant communication. All participants noted that the communication related Agile practices like osmotic communication indeed increased trust between the team members within one location, but they did not seem to have any impact on the trust between two dispersed parts of the team. Some participants noted that the communication practices as proposed by Agile methods could not be applied to the distributed environment – they had to be adjusted to suit the distributed teams, otherwise they did not bring any value. One such example was the direct personal and targeted communication which was simply not possible between team members in both locations. Additionally, the time window for direct communication via phone or video conference was limited to 2 hours per day due to the time difference between two locations (4 PM – 6 PM Polish time, 9 AM – 11 AM American time). Although practices such as kick-off meetings for all team members or exchanging workers for the first few weeks of the project are suggested by some authors to be key success factors in distributed projects [1], some members of the team being described did not even have the chance to see each other in person – they knew each other from remote communication.

It quickly became obvious that in order for the communication to be effective, some changes had to be applied to the processes. The adjustments were proposed collectively by the team members. Those amendments that might introduce risk or have impact on the project quality or schedule had to be approved by the project stakeholders. Once they were, the team might use them in the way they found useful (one of such practices was introducing software for sharing desktops).

One of the adjustments that was seen as crucial for the communication to be effective in the distributed team was introducing more informal written communication. So, apart from phone calls, team members from both locations used informal emails and instant messaging to convey important information. When they knew they would have a conference call, they created a habit to send unstructured informal notes containing discussion points and goals of the call, so that each participant can stick to those points during the call. The team members also started to share their desktops during the calls, so that the other participants could see what they were talking about without the need for describing all the details.

Another adjustment was to introduce more informal forms of documentation. The team initially introduced a 'meeting notes' template to summarize the arrangements made during conference calls. However, they realized that such an informal documentation might be used for conveying other information in a much quicker way than waiting for the next day for the next communication window. So, the team started to create informal visualizations of the interface, informal architecture stencils, as well as conceptual diagrams of the system components. There was another advantage of the informal documentation, other than no need for waiting till the next day – people realized that when they drew a diagram and briefly described it in an informal document, it was much easier for them to convey important information without facing language issues that they might face during direct communication. The Business Analyst from the project described it in her words as follows:

> *"Very quickly did we realize that it was much easier to create a sort of Functional Requirements Document with some details about user stories and acceptance criteria and send it to the other part of the team*

than it is to discuss these requirements during numerous conference calls"
– Business Analyst

Adjustments made to the communication practices increased the team cohesion and understanding between team member, and thus indirectly helped to increase trust between them.

Continuous integration. The continuous integration was given a serious consideration in the project from the research. Initially two version control systems were used, but they were quickly replaced with one subversion system. The team also used software for automatic compilation and builds. The system was integrated with other components that executed automatic integration tests on a timely fashion.

Case study participants were very keen on emphasizing the positive impact of continuous integration on trust between all team members. One of the developers described the case:

"We made several builds of the software per day. Every time anyone committed any new code to the version control system, the system was built and automatically tested. When I saw a green lamp meaning that the build is successful, I thought 'Someone added something to our system. And that works'. This thought was especially encouraging when I realized that the new commit was done by the guys from the other location. That increased my trust in them and in their work."
– Developer

Collective Code Ownership. The Collective Code Ownership is a feature of multiple Agile methods [5]. Some of them, however, praise individual code ownership [8]. The Collective Code Ownership practice, when used, has a very interesting dependency on trust in the team. Namely, it helps to build trust, but at the same time it requires a specific level of trust to be applied [10]. One of the interviewees confirmed that it was the case in the project under study as well:

"We don't treat code as our possession. Instead, we are all responsible for it. To be honest, I don't know whether this approach [Collective Code Ownership] is the result of us trusting each other or it is rather something that builds trust between us. It's both, I think."
– Developer

The Collective Code Ownership practice also helped the team to cement members from both locations by enabling close co-operation on the components being developed. The team members also noted that if they found a particular part of the code was not of the highest quality, the whole team gathered to discuss the issues and to find a proper solution. Because there was no particular person who could be treated as the code owner, no one was blamed at any time for parts that might have been not up to the standards. Instead, the whole team was working together on improving those parts.

Technical Excellence. The Agile methods encourage developing and maintaining technical excellence among the team members. Several practices may be used to achieve technical excellence, from individual coaching to retrospectives and reflective improvement [2]. The study has shown that technical excellence can be an important factor that determines that the team members trust each other. Having at least one senior professional with great experience who is able to coach others increases trust within the team. This is also the case in distributed teams. When the team members respect their knowledge and skills, they create a foundation for trust within the team:

> *"My team mates' technical knowledge and experience increased my trust in them and, generally, in the project future success."*
> – Business Analyst

The study also showed that the awareness of the level of technical excellence, as well as the level of technical excellence itself, in distributed teams can easily be increased by sharing knowledge between both parts of the team. The team from the study organized 'knowledge sharing' sessions. Everyone could participate in them and everyone could benefit from others' knowledge. These sessions helped to cement the team and helped to increase the technical excellence.

5 Conclusions

During the study we obtained several perspectives on the practices that can be used to build trust in geographically dispersed teams. The study results showed that some Agile practices did not have any impact on trust between team members, whereas some could be crucial to the team cohesion and the level of trust in the team. The study also showed that some Agile practices need to be amended to fit distributed teams and that only after the adjustments have been made, can these practices help to build trust between the team members.

The study showed that not only soft practices help to build trust. Also, such technical practices like continuous integration and collective code ownership were crucial in terms of building trust between the team members. Although they might be harder to implement in geographically dispersed teams than in co-located ones [2], implementing them definitely pays off in terms of trust and team cohesion.

The adjustments made to the Agile practices in the project under study definitely helped to create an environment full of trust. The needs for those adjustments came from the team members who best felt what was working well and what did not. They felt that they should trust each other and tried to come up with adjustments that would help them to accomplish that.

We believe that our project shows that the Agile practices can be used to increase trust in distributed teams. Sometimes we may need to introduce some amendments. We may also have to spend more time on choosing the right practices, as some of them not only will not work in dispersed teams, but can be dangerous to the team cohesion.

References

1. Carmel, E., Tjia, P. Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce. Cambridge University Press, Cambridge (2006).
2. Cockburn, A. Crystal Clear: A Human-Powered Methodology for Small Teams. Addison-Wesley Professional, 2004.
3. Derby, E., Larsen, D. *Agile retrospectives. Making good teams great*. Pragmatic Bookshelf (2006).
4. Gerring, J. Case Study Research: Principles and Practices. Cambridge University Press, Cambridge (2007).
5. Highsmith, J. Agile software development ecosystems. Pearson Education, Boston (2002).
6. Miller, A. Distributed Agile Development at Microsoft - patterns and practices. Patterns and practices series. Microsoft (2008).
7. Myers, M. and Young, L. Hidden agendas, power and managerial assumptions in information systems development: an ethnographic case study. Information Technology and People, 10 (3). pp. 224 - 240.
8. Palmer, S. R., Felsing, J. M. A practical guide to Feature-Driven Development. Prentice Hall PTR, Upper Saddle River (2002).
9. Sepulveda, C. Agile Development and Remote Teams: Learning to Love the Phone, Proceedings of the Agile 2003 Conference (2003).
10. Shore, J., Warden, S. The art of Agile Development. O'Reilly Media, Inc., Sebastopol (2007).
11. Skalbania, J. How do agile soft and technical practices help to maintain trust and team cohesion in distributed software projects? M.Sc. project, University of Oxford, Oxford (2009).
12. Venolia, G. et al. Cooperative and human aspects of software engineering. 30th International Conference on Software Engineering (ICSE 2008), Leipzig, Germany, May 10-18, 2008, Companion Volume 2008, Leipzig (2008).