Using Agile Practices to Build Trust in an Agile Team: A Case Study

Abstract: Trust is an important aspect of any software development team, but particularly with self-managing teams as team members are very dependent on one another. Agile teams are considered to be self-managing and they employ many different agile practices to function as an agile team. While there have been many studies of trust in software development teams few have examined trust in an agile context with even less focus on how specific agile practices may contribute to trust. The purpose of this study is to examine how three agile practices - the daily stand-up, iteration planning and iteration retrospective - may support and facilitate trust in an agile team. An exploratory case study of one agile team was conducted. The findings indicate that while factors such as environmental conditions and personal characteristics of team members must be considered, agile practices can also contribute to building trust among team members. They may also highlight the existence of a lack of trust.

1 Introduction

Agile software development (ASD) refers to a group of agile methodologies that focus on developing software in short time periods (iterations). They allow requirements to evolve and change during iterations, encourage close collaboration between agile teams and users, and have teams that are self-organising and cross-functional (AgileAlliance, 2001). ASD has evolved since the mid-1990's and there are now many different agile methodologies in existence such as eXtreme Programming (XP), Scrum, Dynamic Systems Development Method (DSDM) and Feature Driven Development (FDD). These methodologies are often called "lightweight" methodologies as they differ in their approach to the traditional predictable "plan-driven" method of developing software, which requires software teams to follow many processes, and to produce lots of documentation (Boehm, 2002).

The Agile Manifesto places great emphasis on the agile team and the role of the individuals within the team. Teams should be self-organising and self-managing, contain motivated individuals, be provided with the environment and support they need, and be trusted to get the job done (AgileAlliance, 2001). With ASD the team is

provided with substantially more control than it would have had when using a plan-driven approach to software development. This is a dramatic change for the project manager, who has traditionally been the primary controller (Nerur, Mahapatra and Mangalara, 2005). Project managers now need to place great trust in their team members to make the right decisions and complete their tasks in a timely manner. One way of ensuring that this may occur is through the various agile practices that are used by the agile team (see section 2.3 for further detail on agile practices).

1.1 Research Objective and Motivation

Research in the area of ASD has grown in recent years due to the increase in the number of software project teams that use an agile methodology (Abrahamsson, Conboy and Wang, 2009; Conboy, 2009; McEvoy and Butler, 2009). The objective of this study is to explore how trust amongst agile team members can develop and be nurtured as a result of using agile practices, which in turn may develop positive team outcomes such as fostering better relationships/cohesiveness amongst team members or improved team performance. Previous studies highlighted the importance of trust in agile teams (Das and Teng, 2001; Mayer, Davis and Schoorman, 1995; Nerur et al., 2005), but little has been said about how the use of agile practices can increase or decrease trust among team members, which is a motivation for this research. There have also been recent calls for further research that is more practice-focused (Dybå and Dingsøyr, 2008) and to investigate how each distinct agile practice can help to optimise the performance of an ASD team (Maruping, Venkatesh and Agarwal, 2009). Consequently, three practices were selected for the purposes of this study (see Table 1), on the basis that they are amongst the more commonly used agile practices by practitioners (VersionOne, 2009). Each of these practices is related to the management and control of an agile project and requires the collective participation of all team members with a focus on people, communication, interaction and teamwork.

Table 1. Agile practices studied (Beck and Andres, 2005; Elssamadisy, 2008; Schwaber and Beedle, 2002)

Agile Practice	Description
Daily Stand-Up	The daily stand-up is a short daily status team meeting lasting a maximum of 10-15 minutes typically conducted at the same time each day. The meeting is conducted with team members standing up. During the meeting team members explain briefly what they accomplished since the previous meeting, what will be completed by the next meeting and indicate any impediments that may prevent them from completing these tasks.
Iteration Planning	The iteration planning session is a meeting that takes place at the start of each iteration where the team collectively define and plan tasks that must be completed during the next iteration.
Iteration Retrospective	An iteration retrospective is a meeting that is held at the end of each iteration where the project team reflects on what went well in the iteration, what did not, and what could be improved for future iterations.

The remainder of this paper is structured as follows. Section 2 provides an overview of the literature on teams, agile teams, agile practices and trust and then intro-

duces the research question. Section 3 provides details on the case organisation. Section 4 details the methodological approach for this study. Section 5 presents the findings from the case study and the final section discusses the findings, details the limitations of the research and outlines recommendations for further research.

2 Background

2.1 Teams

Teams are groups of individuals that work together, are dependent upon one another and have one or more tasks to perform in order to accomplish various goals (Hackman, 1990; Mayer et al., 1995). Teams should comprise of individuals who are technically competent, are productive, committed to the team, and have good problem solving and interpersonal skills (Jurison, 1999). There is also value in ensuring that a team has a mix of personality types, both introvert and extrovert, which can lead to a more successful team (Jurison, 1999). There are many conditions that must be met in order for teams to be effective such as: creating a team that can work well together; ensuring the team are committed to the organisation; providing the team with autonomy to make decisions; and creating a supportive environment that provides the team with all the necessary resources and skills in order for them to conduct their work (Wageman, 1997; Wageman, Fisher and Hackman, 2009).

Teams can be manager-led or they can be self-governing and self-managing (Hackman, 1990). Self-governing teams set their own goals, select new members, and manage and execute work of their own design (Hackman, 1990). Self-managing teams are teams that have responsibility for managing their own work and behaviours but, others usually make decisions about goals, team structure, and organisational supports (Barker, 1993; Cohen, Chang and Ledford, 1997; Manz and Sims, 1987). Both types of teams are empowered and have autonomy to make decisions about their tasks and the processes that they use, which are traditionally the responsibility of supervisors and managers (Alper, Tjosvold and Law, 1998; Cummings, 1978). To perform well as a team all members must be committed to the team and must feel that they have the support of other members (Bishop, Scott and Burroughs, 2000) as the relationship between individuals within teams can impact on the dynamics of the team (Gruenfeld, Mannix, Williams and Neale, 1996). For example, teams of individuals that are more familiar with each other may be more effective at sharing information and views than those who are not (Gruenfeld et al., 1996).

2.2 Agile Teams

Agile teams are considered self-managing and self-governing (Cockburn and Highsmith, 2001). Yet, it cannot be assumed that by putting a group of individuals together in a team and calling them ‘self-managing’ means they are automatically agile (Moe, Dingsøyr and Dybå, 2010). While the optimal size of an agile team has been debated, ASD teams are typically small with no more than ten team members (Schwaber and Beedle, 2002). Team members should have a range of skills, be

cross-functional and have the ability to complete the required tasks (Elssamadisy, 2008, p128). A team must be empowered to make decisions and is responsible for meeting the goals of each iteration in whatever way it deems appropriate (Schwaber and Beedle, 2002). However, the team must conform to any existing standards within the organisation such as coding standards, hardware/software platforms etc. (Schwaber and Beedle, 2002).

To ensure a team produces quality work an appropriate and supportive environment must be available to team members, for example, ensuring availability of required tools, and open-office space to facilitate open communication. There is also a necessity for team members to be cooperative, collaborative, trusting, have good relationships with each other, and be able to make decisions quickly (Cockburn and Highsmith, 2001). It has been questioned whether agile methodologies are suitable for a distributed team for many reasons including the possibility that distributed team members are less likely to feel part of the same team as co-located team members (Ramesh, Cao and Mohan, 2006). These can be alleviated somewhat by site visits, the facilitation of collaboration and knowledge sharing and supplementing informal communication with documentation (Ramesh et al., 2006).

2.3 Agile Practices

Each agile methodology details various practices that distinguish it from other agile methodologies, but they each follow the same underlying agile principles (AgileAlliance, 2001) where a practice can be described as a "common way of acting", which is accepted by a group of individuals as the "correct way to do things" (Hansson, Dittrich, Gustafsson and Zarnak, 2006). Agile teams can choose to adopt the agile practices that suit their environment or that work well for them, bearing in mind that these practices may span several agile methodologies (Elssamadisy, 2007; Hansson et al., 2006). These agile practices may be technical (e.g. test driven development, continuous integration), relate to planning (e.g. iteration planning, daily stand-up), or could relate to the agile environment (e.g. co-located team, self-organising team). The main premise of ASD, regardless of the agile methodology that is used, is to deliver value to the customer as early as possible by reducing the time to market, increasing quality, increasing flexibility and providing the ability to respond to change and reduce costs (Boehm and Turner, 2003; Elssamadisy, 2007). Many software teams adopt agile practices for some or all of these reasons:- whether they are successful or not depends on the agile practices adopted, the team, and the individuals within the team (Elssamadisy, 2008).

2.4 Trust

The concept of trust has been studied in many different contexts, yet there is little agreement on a single definition with the term used in many different ways (Blomqvist, 1997; Kramer, 1999; Lewicki, McAllister and Bies, 1998; McKnight, Cummings and Chervany, 1998; Rousseau, Sitkin, Burt and Camerer, 1998). Trust, or a lack of trust, can exist between individuals, groups and organisations (Das and Teng, 2001) with trust fostering cooperation amongst parties (Rousseau et al., 1998).

As organisational teams become more diverse with team members from a variety of backgrounds and culture, the development of trust between all members is extremely important for them to work together effectively (Mayer et al., 1995). Individuals with different personality types, experiences and cultural backgrounds vary in propensity in how likely they are to trust others (Hofstede, 1980) with levels of trust evolving or diminishing over time as they interact with each other and observe each other (Das and Teng, 2001; Mayer et al., 1995). Distributed teams face other challenges such as lack of control, lack of cultural understanding, miscommunication, limited opportunity to communicate orally due to time differences and lack of team morale and trust between team members (Ramesh et al., 2006).

The emergence of self-managing agile teams increases the importance of trust among team members as members are relatively free to develop the processes they prefer and to set targets they consider appropriate (Das and Teng, 2001; Mayer et al., 1995). Team members that collaborate and trust each other are imperative for the success of an agile project, which may be difficult for developers who are used to working predominantly on their own (Nerur et al., 2005). Individuals or teams must believe that each individual within the team has the ability, knowledge, and competence to complete the tasks required and they must also have high personal and moral integrity (Mayer et al., 1995). Therefore, it is important to maintain and strengthen trust between team members. It may take some time and effort for an organisation to build a culture of trust amongst team members (Nerur et al., 2005), but it is possible that this may be facilitated and supported by the use of agile practices. Prior research in this important area is limited, which leads us to the following research question:

How do agile practices contribute to building trust in an agile software development team?

To answer this research question (see research model in Fig. 1) a case study approach is used where a single agile team is studied. This research is part of ongoing research for a Ph.D. and to date data has been collected from one team. At this point this research is exploratory and data will be collected from several other teams in the near future.

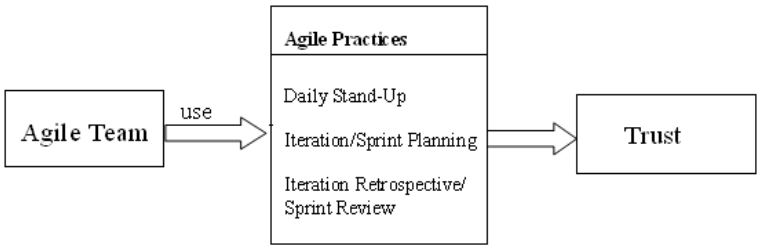

Fig. 1. Research model

3 The Case Organisation Studied

The organisation selected for this case study is a large multinational financial services organisation with offices located worldwide. A decision was made by the Head Office in the United States to introduce a customised version of an agile methodology across the organisation. The Research & Development (R&D) division in Ireland was the first within the organisation to pilot the use of an agile methodology for developing software. The project studied is a long-term project that was in existence for two years at the time of the study and it is envisaged to continue for at least another year. The project involves the development of a set of back-end Web services that are used by various front-end applications for developing financial analysis documents. The end users are financial analysts across several business units, although the direct customers are the IT groups in six different business units who develop the front-end applications. This results in a number of different customers, all whom have competing needs.

The team composition has changed intermittently since its inception. At the time of data collection, the team was composed of eleven individuals distributed between the United States, Ireland and India. The development team was primarily based in Ireland with the Quality Assurance (QA) function based in India and a database specialist and customers based in the United States. The customer was not directly involved in the project on a day-to-day basis with the analyst acting as the "proxy customer". Shortly after data collection commenced, the two QA team members based in India, who were both testers, departed from the team. Recruitment of two replacement team members is currently taking place, but these will be now based in Ireland. The team currently consists of a project manager, developers, analyst, and a database specialist. All but one of the team is very experienced, with most of their experience obtained in a non-agile environment. A profile of the current team is detailed in Table 2.

Table 2. Team Profile

Role	Software development experience	Experience in organisation	Experience in the agile team	Location
Project Manager [P1]	13 years	8 years	3 years	Ireland
Analyst [A1]	15 years	2.5 years	2 years	Ireland
Database Specialist [DB1]	15 years	5 years	11 months	USA
Developer [D1]	10 years	3 years	3 years	Ireland
Developer [D2]	10 years	3 years	6 months	Ireland
Developer [D3]	12.5 years	2.5 years	1.5 years	Ireland
Developer [D4]	2.5 years	2.5 years	2.5 years	Ireland
Developer [D5]	10.5 years	4.5 years	1 year	Ireland

Three week iterations are used by this agile project team. The iteration planning meeting and iteration retrospective meetings are generally combined into one meeting (approximately one hour in duration) with distributed team members available on a conference call. The first 15 minutes of this meeting consists of the iteration retrospective. The project manager asks team members in turn to briefly comment on their work in the previous iteration, to indicate what went well and what can be im-

proved for future iterations. The remainder of the meeting focuses on iteration planning where tasks (user stories) for the next iteration are agreed upon, and estimates are reviewed. Team members are aware of the user stories that are assigned to them prior to the meeting and each team member will have prepared a time estimate for each task.

Daily stand-ups are held four days a week as on the fifth day a project team meeting is held instead of the daily stand-up meeting. Daily stand-up meetings generally last 10-15 minutes and take place in an office as this facilitates a conference call with distributed team members. As with the iteration retrospective, the project manager, or senior developer, if the project manager is not present, directs the meeting. All team members in turn briefly comment on what has been completed since the previous daily stand-up, what they are currently working on and whether there are any 'blockages' inhibiting the completion of a task. In the event that further discussion is required in relation to a task(s) this will typically take place amongst affected team members following the completion of the meeting.

4 Research Design

Case studies are particularly appropriate for exploratory research which is at an early stage of maturity (Benbasat, Goldstein and Mead, 1987). Access to the team in question was readily available, and it was felt that the opportunity should be utilized to conduct a qualitative study and gain an understanding of the agile practices in more detail. An interview guide was developed from the literature on teams, agile methodologies, and trust. It predominantly contained open-ended questions as this provided the researchers with the opportunity to ask additional questions (Cooper and Schindler, 2001). Sample interview questions are available in the Appendix.

Data collection took place over a four month period. Each interview followed a similar structure. Details on the project and on the number and type of agile practices utilized by the team were gathered from the first interviewee only. It was felt that it was only necessary to ask these questions once and it would minimize the amount of time required to interview the remaining participants. All interviewees provided details on their background, and level of experience, and this was then followed with a number of questions under different headings that were asked to each interviewee.

Semi-structured interviews were used to capture the responses of each of the eight participants. This gave the interview a structure, but also allowed thoughts and personal experiences to emerge from participants. Seven of the eight interviews conducted were face-to-face interviews. The remaining interview was conducted using a conference call as this individual was based in the United States. The two individuals in India were part of the team when a number of interviews were conducted, so references are made to these in the findings even though these team members were not interviewed. All interviews were conducted at the offices of the organisation. The interviews lasted approximately one hour each. All interviews were recorded and transcribed. The transcriptions were reviewed, coded and categorised based on the interview guide. The categories were then sub-divided into further categories in order to identify patterns and themes and to validate the data from different

individuals (Miles and Huberman, 1999). The findings were analysed and validated by cross-checking the findings with each of the other participants.

5 Findings

The findings from the interviews conducted are presented below. The team studied is predominantly a well-established, experienced, self-organising team with team members appearing to have a good work ethic and track record of delivering on what had been promised. Team members are very collegiate and supportive of each other and appear to work together as a unit with many items discussed collectively. The team "*look out for each other and make sure that when something goes wrong…it is very much a team effort to fix it [D5]*". This team appears to be very driven and members are constantly looking for more work when their own tasks are completed. This may stem from the current working environment where there were redundancies a number of months ago and each team member has a desire to "*be busy as the last thing they want is having no work [D1]*".

5.1 Working Environment

The current working environment is considered open and supportive by all. The iteration planning and retrospective meetings are an open forum for knowledge sharing and feedback where "*the information sharing is important. It's good to know you can be frank, throw your ideas out there [A1]*". They also help to build trust amongst team members because "*they are having it [meetings] on a more regular basis [P1]*" and "*people get more comfortable over time [D5]*". Individuals are encouraged to voice their opinions in all three meetings without fear of repercussions and no-one has ever been "*been reproached for expressing an opinion [D4]*". If a task takes longer than estimated the individual is not reprimanded, nor looked on negatively, but instead is asked "*did we do the estimate wrong, or have you got something that is blocking you? [A1]"* and is helped to complete the task. If the environment was not as supportive there may be a tendency for individuals to "*become more conservative when [they] plan [D4]*" so as to avoid negative repercussions, which could "*be very detrimental [D4]"* to the project.

The team in Ireland is co-located with team members sitting in very close proximity of each other. This facilitates face-to-face communication between co-located team members and it is not uncommon to "*see someone over at another person's desk and then drawing something out on paper to explain a certain piece [D4]*". One developer [D3] believes that co-location allows trust to develop amongst team members as "*he can hear everything that is going on*" between team members and can contribute to conversations if necessary.

5.2 Building Relationships

The interaction at each of the meetings (iteration planning, iteration retrospective and daily stand-up) helps to build a rapport between team members. There is usually

some banter between the different sites at the start of a meeting which "*probably helps you to develop a relationship there that you are not really aware of [D3]*". It also "*sets a good tone [D4]*" for the meeting. The meetings provide team members with an "*opportunity to question and once you get a valid answer back…, well then that does help [A1]"*. As the team is distributed across three different cultures it can be difficult for team members to build good relationships with other distributed team members, especially when "*you haven't met face-to-face [D4]*". Face-to-face communication between team members in each of the locations is limited with travel generally only taking place between Ireland and the United States. For example, *"developers go over there [United States] for about a week each. It is usually only kind of when it is close to production time and we are kind of ramping up and we need to be on site just in case any issues pop-up [D1]"*. One site visit has taken place between India and Ireland where the QA Manager from India travelled to Ireland. But, the main involvement of this individual related to "*deciding where the [QA] resources go [D1]*" as opposed to contributing to the daily workload.

All of these agile practices have resulted in the development and fostering of relationships between co-located and distributed team members that may not have existed otherwise. The project manager is of the opinion that "*anything that encourages conversation between people is going to build up a level of trust between developers [P1]*". The retrospectives "*work over time… and help with understanding people… as the more that is spoken frequently, …a bond kind of builds up [D5]*". There is a "*good culture of just picking up the phone [D4]*" to ask another team member a question. The agile practices have helped distributed team members to feel part of the team "*because of the continuous communication between the team, it helped me feel part of the team [DB1]*". New team members also integrated faster into the team and quickly built relationships with other team members because they were required to participate in the daily stand-ups iteration planning and retrospectives from the outset. This is particularly important as difficulties have been experienced with the QA team in India where "*they chop and change them [team members] regularly [D2]"* resulting in the regular re-building of relationships.

Team members do not consider the customer part of the team and there is an apparent lack of trust between the customer and the team. "*Response times from the customer are very slow… it can be hard to get their time…there can be misunderstandings…they have their own agenda [D2]*". The customer does not participate in any of the meetings and it can "*get to the stage where you [the team] may have a chat with the customer and they would ask you to do X, Y and Z… and we get them to send an email so we have it in writing [D3]*". The customer is also slow to review releases of the software and "*it's frustrating to go back there [to the customer] if there is nothing happening [A1]*". One developer is of the opinion that even if the customer did participate in the daily stand-up and the retrospectives "*they wouldn't be able to contribute…. as the project is very technical [D1]*". However, a site visit by a team member to the United States "*was very beneficial… as it encouraged a lot more conversation [with the customer] [D1]*".

One team member (D3) does not believe that any of the agile practices contribute to building trust and considers them a chore which often disrupts his working day. Another team member (D4) believes that they are routine where team members

speak in turn "*even though you [team members] have nothing to say really*". Yet, they are still considered "*a crucial part of the [development] process [D3]*".

5.3 Communication

The daily stand-up meetings in particular are a daily touch-point for all team members. They force team members (co-located and distributed) to meet and communicate with each other on a daily basis and "*keep the lines of communication open [D4]*". Speaking to each other on such a regular basis improves communication and helps individuals to better understand each other, become familiar with their personalities and traits and be more comfortable in their interactions with each other. One developer indicated that *"I've worked on teams where you'd hardly speak to each other all day [D2]"* which was echoed by other developers (D1, D3). Even though the daily stand-up was considered "*tedious*" at times the team members welcome the opportunity to speak to other team members on a daily basis, even if the conversation "*drifts off into off topic stuff ... it is useful to hear what they're [other team members are] doing [D2]*". They also encourage more informal communication (Instant Messenger, phonecalls, and ad-hoc conversations in the office) amongst team members outside of meetings and team members do not feel that they need to wait for the daily stand-up to take place in order to discuss a problem.

5.4 Trust amongst Team Members

A lot of trust appears to exist between team members. The project manager trusts team members to "*work a good solid day [D1]*". He also trusts the team to accurately define estimates for tasks as "*they [developers] will have more context than me [P1]*" and to then deliver on those estimates. It is rare for other team members or the project manager to question a time estimate: "*planning estimates aren't really collective decision they're just presented by developers as their individual times and agreed [D1]"*. This was corroborated by the project manager who "*doesn't tend to take a lot of decisions [P1]*". The daily stand-ups in particular help the project manager to keep track of tasks assigned to all team members (both local and distributed) and how they are progressing.

The agile practices have helped to alleviate the possibility of distrust that can be experienced with distributed team members from different cultures. Participation by the QA (Indian) team in daily meetings and the planning/retrospective meetings helps to build trust with them as the QA team have a tendency to refrain from being too vocal and are "*fairly timid kind of guys, they don't really say much other than 'this is what I did' and 'this is what I'm going doing today' [D2]*". This may be "*a cultural thing or may be the lack of experience in the team [D2]*". This is of particular importance to the project manager who has had some trust concerns with the distributed team, but the "*stand-up is a great way to keep on top of it [progress] [P1]*". The team "*know very quickly [P1]"* of any actual or potential delays, which can be addressed immediately."*There is a lot of conversation going on that wouldn't happen if these practices weren't being used [P1]"*.

All team members believe that their colleagues are competent and can complete the tasks allocated to them. They trust and accept that their colleagues are honest when determining estimates and can be believed when they say that a task is complete. Yet, at the same time no-one will "*go to them [a developer] tomorrow and ask them if they had it [a task] done [D3]*".

6 Discussion and Further Research

These findings contribute to the literature on trust and agile teams by attempting to provide some insight into how daily stand-up meetings, iteration planning and iteration retrospectives contribute to trust in an agile team. Overall, the team studied have reported a predominantly positive view of these three agile practices. While the agile practices are not the only factors that help to increase trust in the team, they are acknowledged to be a contributing factor. What is particularly important is that they can also highlight where there is a lack of trust between members, which can be addressed by the project manager or the team.

The literature has shown that it is important for an organisation to build a culture of trust among team members (Nerur et al., 2005). This team trusts each other very much, which may be due to a number of different factors. The team studied was very experienced and cohesive. As a consequence of using an agile methodology the team had autonomy to make their own decisions, set their own deadlines for an iteration, and could control what they do within the team which were defined during the iteration planning meeting. The project manager listened to the team, did not appear to micromanage and generally supported the decisions made by the team. The environment itself was very supportive and using these agile practices has provided an opportunity for trust to foster and develop among team members.

Wageman, Fisher and Hackman (2009) believe that regular interventions can help team members resolve difficulties with a task much more quickly, or can avoid work on a trivial or non-essential task, increasing the likelihood that the team will remain focused on relevant work and as a result, be more productive. In ASD teams, such interventions take place on a daily basis through the daily stand-up meeting where current concerns can be raised and discussed, ideas and problems can be shared and advice provided in a constructive way. These meetings in addition to the iteration planning and iteration retrospective meetings provide transparency on tasks and whether or not tasks are being completed on time.

In ASD projects the customer is typically part of the team and is ideally involved in the project on a daily basis, which should help to build a relationship and trust between the customer and the team. In this particular project the lack of participation by the customer in the daily stand-up, iteration planning and iteration retrospectives has had an impact on the relationship with the customer. As a result, the team do not place a great deal of trust in the customer. This suggests that lack of participation by any key team member in these meetings can result in the development of a lack of trust.

Distributed software development teams face particular challenges, particularly in relation to culture, and communication when the time zones vary dramatically

(Ramesh et al., 2006) and it is important to find ways to address these problems. One possible way is to use agile practices which provide distributed team members with a facility to communicate and interact with the team on a daily/regular basis and help them to feel part of the team. Daily meetings encourage a certain amount of informal communication, such as social conversations as the start of a meeting, which can contribute to breaking down any cultural barriers that may exist and building a relationship with the team members. This may lead to an increase in the levels of trust between team members as feedback is obtained regularly and team members can form their own opinion as to whether other team members are competent and can be trusted to complete good quality work on time.

People are extremely important in an agile team and it is imperative that they can work together, have good relationships, and trust each other to deliver on what is promised (Nerur et al., 2005). As the demand for successful software development projects continues it is important that trust is a core element of a team. Many different agile practices may contribute to trust amongst team members, but the findings of this study are a first step towards understanding how three different agile practices can support and facilitate trust in an agile team. At the same time it is important to remember that another agile team in a different environment may not use these agile practices in the same way and as a result may not have the same positive experience.

This research is limited by virtue of the fact that a single case study is utilized as the research method. The findings are therefore, only representative of this team. While the team was initially distributed across three continents, team members in one of the locations departed from the team during data collection and were not replaced locally. The perspectives of these team members may have provided different insights into the agile practices used as they were distributed team members of a different culture. A second limitation relates to the number of practices studied. This research only focused on three practices. Future research should examine other agile practices to determine if they too impact on trust amongst agile team members.

Appendix

This appendix details a sample of the interview questions asked to participants as part of this study in relation to trust.

1. Do these agile practices encourage/enable trust among your team members?
2. How do they do this/inhibit/prevent this?
3. Can you provide an example to demonstrate this?
4. Do these agile practices encourage/enable trust with team members that are distributed?
5. How do they do this/inhibit/prevent this?
6. Do these agile practices encourage/discourage individuals to talk freely to other team members about difficulties that they may have in relation to the project?

References

Abrahamsson, P., Conboy, K. and Wang, X. (2009) Lots done, more to do: The current state of agile systems development research, European Journal of Information Systems, 18 (4) 281-284.

AgileAlliance (2001) "Manifesto for Agile Software Development" [Online] Available at: (www.agilemanifesto.org),

Alper, S., Tjosvold, D. and Law, K. S. (1998) Interdependence and Controversy in Group Decision Making: Antecedents to Effective Self-Managing Teams, Organizational Behavior and Human Decision Processes, 74 (1) 33-52.

Barker, J. R. (1993) Tightening the Iron Cage: Concertive Control in Self-Managing Teams, Administrative Science Quarterly, 38 (3) 408-437.

Beck, K. and Andres, C. (2005) Extreme Programming Explained: Embrace Change, 2nd edn, Addison-Wesley, Boston, MA.

Benbasat, I., Goldstein, D. K. and Mead, M. (1987) The Case Research Strategy in Studies of Information Systems, MIS Quarterly, 11 (3) 368-386.

Bishop, J. W., Scott, K. D. and Burroughs, S. M. (2000) Support, commitment, and employee outcomes in a team environment, Journal of Management, 26 (6) 1113-1132.

Blomqvist, K. (1997) The many faces of trust, Scandinavian Journal of Management, 13 (3) 271-286.

Boehm, B. (2002) Get Ready for Agile Methods, With Care, IEEE Computer, 35 (1) 64-69.

Boehm, B. and Turner, R. (2003) Using risk to balance agile and plan-driven methods, Computer, 36 (6) 57-66.

Cockburn, A. and Highsmith, J. (2001) Agile Software Development: The People Factor, IEEE Computer, 34 (11) 131-133.

Cohen, S. G., Chang, L. E. I. and Ledford, G. E. J. (1997) A Hierarchical Construct of Self-Management Leadership and its Relationship to Quality of Work Life and Perceived Work Group Effectiveness, Personnel Psychology, 50 (2) 275-308.

Conboy, K. (2009) Agility from first principles: Reconstructing the concept of agility in information systems development, Information Systems Research, 20 (3) 329-354.

Cooper, D. R. and Schindler, P. S. (2001) Business Research Methods, 8th edn, McGraw Hill, London, England.

Cummings, T. G. (1978) Self-Regulating Work Groups: A Socio-Technical Synthesis, The Academy of Management Review, 3 (3) 625-634.

Das, T. K. and Teng, B.-S. (2001) Trust, Control, and Risk in Strategic Alliances: An Integrated Framework, Organization Studies, 22 (2) 251-283.

Dybå, T. and Dingsøyr, T. (2008) Empirical Studies of Agile Software Development: A Systematic Review, Information and Software Technology, 50 (9-10) 833-859.

Elssamadisy, A. (2007) Patterns of Agile Practice Adoption: The Technical Cluster, C4Media, USA.

Elssamadisy, A. (2008) Agile Adoption Patterns: A Roadmap to Organizational Success, Addisson Wesley, USA.

Gruenfeld, D. H., Mannix, E. A., Williams, K. Y. and Neale, M. A. (1996) Group Composition and Decision Making: How Member Familiarity and Information Distribution Affect Process and Performance, Organizational Behavior and Human Decision Processes, 67 (1) 1-15.

Hackman, J. R. (1990) Groups that Work (and those that don't). Creating Conditions for Effective Teamwork, Jossey-Bass Inc. San Francisco, CA.

Hansson, C., Dittrich, Y., Gustafsson, B. and Zarnak, S. (2006) How Agile are Industrial Software Development Practices?, Journal of Systems and Software, 79 (9) 1295-1311.

Hofstede, G. (1980) Motivation, Leadership, and Organization: Do American Theories Apply Abroad?, Organizational Dynamics, Summer, p42-63.
Jurison, J. (1999) Software Project Management: The Manager's View, Communications of the AIS, 2 (17) 1-50.
Kramer, R. M. (1999) Trust and distrust in organizations: Emerging perspectives, enduring questions, Annual Journal of Psychology, 50 (1) 569-598.
Lewicki, R. J., McAllister, D., J. and Bies, R. J. (1998) Trust and Distrust: New Relationships and Realities, Academy of Management Review, 23 (3) 438-458.
Manz, C. C. and Sims, H. P., Jr. (1987) Leading Workers to Lead Themselves: The External Leadership of Self- Managing Work Teams, Administrative Science Quarterly, 32 (1) 106-129.
Maruping, L. M., Venkatesh, V. and Agarwal, R. (2009) A Control Theory Perspective on Agile Methodology Use and Changing User Requirements, Information Systems Research, isre.1090.0238.
Mayer, R. C., Davis, J. H. and Schoorman, F. D. (1995) An Integrative Model of Organizational Trust, Academy of Management Review, 20 (3) 709-734.
McEvoy, J. and Butler, T. (2009) The role of project management in ineffective decision making within Agile software development projects, European Journal of Information Systems, 18 (4) 372-383.
McKnight, D. H., Cummings, T. G. and Chervany, N. L. (1998) Initial Trust Formation in New Organizational Relationships, Academy of Management Review, 23 (3) 473-490.
Miles, M. and Huberman, A. (1999) Qualitative Data Analysis, Sage, London.
Moe, N. B., Dingsøyr, T. and Dybå, T. (2010) A teamwork model for understanding an agile team: A case study of a Scrum project, Information and Software Technology, 52 (5) 480-491.
Nerur, S., Mahapatra, R. and Mangalara, G. (2005) Challenges of Migrating to Agile Methodologies, Communications of the ACM, 48 (5) 72-78.
Ramesh, B., Cao, L. and Mohan, K. (2006) Can Distributed Software Development be agile?, Communications of the ACM, 49 (10) 41-46.
Rousseau, D. M., Sitkin, S. B., Burt, R. S. and Camerer, C. (1998) Not so different after all: A Cross-Discipline View of Trust, Academy of Management Review, 23 (3) 393-404.
Schwaber, K. and Beedle, M. (2002) Agile Software Development with Scrum, Prentice Hall, NJ, USA.
VersionOne (2009) State of Agile Development Survey 2009 (Accessed: 31 March 2010) [Online]. Available at http://pm.versionone.com/StateofAgileSurvey.html
Wageman, R. (1997) Critical Success Factors for Creating Superb Self-Managing Teams, Organizational Dynamics, 26 (1) 49-61.
Wageman, R., Fisher, C., M. and Hackman, J. R. (2009) Leading Teams when the Time is Right: Finding the Best Moments to Act, Organizational Dynamics, 38 (3) 192-203.