Technical Report 12-02:
Understanding Team Dynamics in
Distributed Agile Software Development

Siva Dorairaj, James Noble and Petra Malik

January 16, 2012

Abstract

Team dynamics are patterns of interaction among team members that determine the performance of the team. Success of Agile software development depends on team interaction. Team interactions are, however, affected in distributed teams. We conducted a Grounded Theory study to investigate team interaction in distributed Agile teams involving 40 Agile practitioners from 24 different software companies in the USA, Western Europe, Australia and India. We present six eminent strategies that promote team interaction in distributed Agile teams.

1   INTRODUCTION

Team dynamics in software development are patterns of interaction among team members that determine the performance of the team [1]. Several studies assert that team dynamics are important characteristic of high performance teams [2, 3]. Effective team interaction provides avenues for team members to state ideas and opinions without barriers, listen actively to understand the concerns of other team members, and provide timely suggestions to the problems faced by the team [4, 5]. Success of Agile software development depends significantly on team interaction [6, 7, 8, 9, 10].

Agile teams in distributed software development interact over time and space through technology-mediated communication such as telephone and e-mail [2, 11, 12]. Non-verbal communication such as facial expression and hand gestures that are often missing in technology-mediated communication, decreases the awareness of team member actions [2]. Team interactions are affected in distributed Agile teams [2, 3, 13].This raises a critical question: *How do Agile teams promote team interaction in distributed software development?* We found the answer to this question through a Grounded Theory study that involved 40 Agile practitioners from 24 different software companies in the USA, Western Europe, Australia and India. We found the six eminent strategies that promote team interaction in distributed Agile teams: *'one team' mindset*, *personal touch*, *open communication*, *team collocation*, *team ambassadors*, and *coach travels*.

2 BACKGROUND

Agile methods are a family of software development methods that follow an iterative and incremental style of development, designed to deliver quality software on time and on budget [14]. Agile methods include XP (eXtreme Programming) [15], Scrum [16], ASD (Adaptive Software Development) [17], DSDM (Dynamic Systems Development Method) [18], FDD (Feature Driven Development) [19, 20], and Crystal Clear [21]. Scrum and XP are considered to be the most widely adopted Agile methods in software development projects [22]. The Agile Manifesto [23] states the four values of Agile methods:

Individuals and interactions over processess and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right,
we value the items on the left more.

Agile methods promote continuous face-to-face interaction between team members to fosters teamwork and build trust in the team [24]. One of the principles behind the Agile Manifesto states that an Agile team reflects on its team interaction at regular intervals, and adjusts its behaviour to become more effective [23]. Agile software development relies on teamwork, and the practices characterized by team interaction, such as daily standup meetings and pair-programming, are critical for the success of Agile projects [25]. Inability to meet face-to-face on a daily basis and lack of team interaction among distributed team members affect the team performance [26]. Agile teams in distributed software development, in the absence of collocation, interact over time and space through technology-mediated communication such as telephone, video conferencing, or electronic mail [2]. As non-verbal communication such as facial expression and hand gestures are often missing in technology-mediated communication, distributed teams experience decreased awareness of team member actions [2]. Team interaction, however, can be difficult or significantly limited when teams are separated across several time zones as realtime availability of team members are affected [11, 12].

3 RESEARCH METHOD

3.1 Grounded Theory

Grounded Theory (GT) is a systematic research method that emphasises the generation of theory derived from systematic and rigourous analysis of data. GT was originally developed by Barney G. Glaser and Anslem L. Strauss [27]. 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, and the literature on distributed Agile software development, particularly on team dynamics in distributed teams, is still scarce [6, 28]. Secondly, GT allows researchers to study social interactions and the behaviour of people in the context of solving problems, and Agile methods focus on people and their interactions in software development teams [16]. Notably, GT is increasingly being used successfully to study the social nature of Agile teams [29, 30, 31, 32, 33].

3.2 Data Collection

Our data collection technique was interviewing Agile practitioners. We conducted face-to-face, one-on-one interviews with our participants using open-ended questions that focused on the challenges faced by distributed teams, and the strategies to overcome them. We phrased our questions cautiously so that the issues in distributed Agile software development would emerge from the participants rather than from our own agenda. The interviews were voice-recorded with the consent from the participants. Voice recording the interviews helped us to concentrate on the conversation and to understand participant's main concerns in distributed Agile projects. We often adjusted our interviews to focus on these concerns rather than our prepared questions.

3.3 Participants

We interviewed 40 Agile practitioners from 24 different software organisations in the USA, Western Europe, Australia and India. Participants adopted Agile methods, primarily Scrum and XP, in their distributed software development projects. We interviewed participants from a range of different roles within the distributed Agile projects: Scrum Masters, Agile Coaches, Developers, Quality Analysts, Business Analysts, and Senior Management (e.g. Vice President, Human Resource Manager, Director of Technology). Table 1 shows participant and project details. The projects were distributed between 2 or 3 countries, project durations varied from 6 to 24 months, team size varied from 8 to 50 people on different projects, and project iteration varied from 2 to 5 weeks. Due to privacy and ethical consideration, we will only identify our participants using the codes P1 to P40.

3.4 Data Analysis

We transcribed the interviews, and used open coding to analyse the interview transcripts line by line [34, 35]. Open coding breaks down, examines, compares, conceptualises, and categorises the data [36]. We 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 compared codes, many fresh concepts emerged. These concepts were themselves analysed using constant comparison method to produce a third level of abstraction called a *category*. Figure 1 shows the concepts *'One Team' Mindset*, *Personal Touch*, *Open Communication*, *Team Collocation*, *Team Ambassadors*, and *Coach Travels* that gave rise to the category *Building Team Dynamics*.

4 RESULTS

In this section we present the strategies that promote team interaction in distributed Agile teams: the category *Building Team Dynamics* and its underlying concepts. We have selected quotations from our interviews to illustrate each concept.

Table 1: Participant and Project Details. (Agile Position: Scrum Master (SM), Agile Coach (AC), Developer (DEV), Business Analyst (BA), Quality Analyst (QA), Senior Management (MGT))

Participant (code)	Agile Position	Project Distribution	Agile Method	Team Size	Project Duration (months)	Iteration (weeks)
P1	DEV	USA-India	Scrum	8 to 10	10	2
P2	AC	USA-India	Scrum & XP	12 to 14	12	2
P3	SM	USA-Western Europe-India	Scrum	10	8	3
P4	AC	USA-China	Scrum & XP	10	8	2
P5	AC	USA-India	Scrum & XP	8	12	2 to 3
P6	DEV	USA-UK	Scrum & XP	20 to 22	8	2
P7	AC	USA-Argentina-India	Scrum & XP	18	6	2
P8	DEV	USA-Australia-India	Scrum & XP	9 to 10	8	2
P9	DEV	Western Europe-Brazil	Scrum & Lean	14	24	2 to 3
P10	SM	USA-Argentina-India	Scrum	10 to 12	8	3
P11	SM	USA-Middle East-India	Scrum & XP	13	10	2
P12	DEV	USA-India	Scrum & XP	12	18	2
P13	SM	USA-India	Scrum & XP	17 to 20	5	2
P14	DEV	USA-India	Scrum & XP	16 to 17	36	2
P15	QA	USA-India	Scrum & XP	16	18	2
P16	SM	USA-India	Scrum & XP	16	18	2
P17	DEV	USA-India	Scrum & XP	16	18	2
P18	BA	UK-India	Scrum & XP	8	12	2
P19	DEV	USA-India	Scrum	8 to 10	10	3
P20	MGT	Australia-India	Scrum & XP	9 to 12	12	2 to 3
P21	SM	USA-Australia	Scrum	15	9	2
P22	SM	Australia-India	Scrum & XP	9 to 12	12	2 to 3
P23	QA	Japan-India	Scrum	7 to 8	4	5
P24	AC	Western Europe-India	Scrum & XP	9	5	2
P25	SM	USA-India	Scrum & XP	24	6	3
P26	AC	USA-India	Scrum & XP	16	ongoing	3
P27	SM	USA-Brazil	Scrum & XP	30	6	2
P28	MGT	USA-India	Scrum	20	18	3
P29	SM	USA-India	Scrum & XP	14	10	2
P30	AC	Western Europe-India	Scrum & XP	8 to 10	ongoing	2 to 3
P31	AC	UK-India	Scrum & XP	15 to 20	ongoing	3
P32	MGT	UK-South Africa	Scrum & XP	12	18	2
P33	AC	Australia-Eastern Europe-India	Scrum & XP	50	24	3
P34	AC	USA-India	Scrum & XP	6 to 8	10	2
P35	AC	USA-India	Scrum & XP	8	18	3
P36	QA	Canada-India	Scrum & XP	10 to 15	18	2
P37	DEV	Western Europe-India	Scrum & XP	16	4	2
P38	BA	USA-India	Scrum & XP	28	ongoing	2
P39	AC	USA-India	Scrum & XP	22 to 25	6 to 7	2
P40	DEV	Australia-India	Scrum & XP	7	6	1
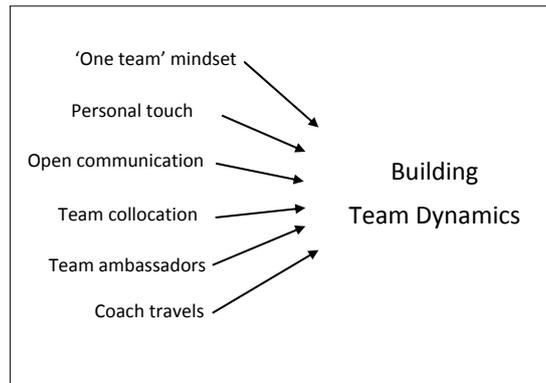
Figure 1: Emergence of category *Building Team Dynamics* from concepts.

4.1   'One Team' Mindset

In Agile software development, team members need to interact frequently with the entire team during meetings, pair-programming, or discussions throughout the project:

> "*Working on a Agile project requires team members to work collaboratively with other people, talking and interacting whether it's in meetings, pairing, [or] talking to people one on one. The social interaction is so important.*" —P20, Management Team.

Crucially, all team members from every location participate in a 'daily stand-up meetings' using technology-mediated communication. Teams prefer video conferencing over telephone conferencing during daily meetings to increase interaction among the team members:

> "*Daily stand-up meetings are mandatory for all team members. We often do Skype video calls. We feel that if we see team members face-to-face, we'll have better interaction with them.*" —P36, Quality Analyst.

Teams understand that the daily standup meetings are important to the entire team, and therefore joint standup meetings are scheduled so that members from all locations are able to participate in the meeting:

> "*When we have a daily standup meeting, we talk about what everyone was doing, we get to know about who is doing what, what are the issues, what happened at the end of our day, and what we need to do for today. We definitely have a joint standup meeting [though] over different time zones, it can be difficult.* " —P11, Scrum Master.

When the team members are separated across several time zones, the daily stand-up meetings with the entire team causes many difficulties for team members. Often, team members from one location have to stay back till late at night, while team members from another location have to come in to work very early in the morning:

> *"We have some people, with abnormal working hours, who come after lunch and work till late night."* —P4, Agile Coach.

Despite the difficulties interacting with one another, distributed teams try to keep a 'one team' mindset to foster their interaction:

> *"We are working as a team in Agile. The team knows that they are not separated just because they are in another building, or another location with some time zone difference. It's only one team. "* —P24, Agile Coach.

In this 'one team' mindset, the project team members and the customer understand that they all belong to one team despite working from different locations:

> *"We have absolutely one team [but] we are working in different locations. And, the client is definitely part of the team."* —P16, Scrum Master.

This 'one team' mindset is the fundamental factor that fosters effective interaction between team members and improve team performance :

> *"We have a one whole team mindset. The team is distributed but it is one whole team. Everyone works as one team, and there is one team performance. "* —P7, Agile Coach.

Participants P1, P3, P7, P16, P18, P20-P29, P32, P33, and P35-P38 explicitly discussed the 'one team' mindset where the members of the distributed team strive to interact and perform as one Agile team.

4.2 Personal Touch

Distributed team members often have difficulties getting to know each other, or even just to 'put a face to the name':

> *"We were working together with Sebastian but we didn't know who is Sebastian. We have not even seen [his] face. It is hard to get the feeling of teamness when you don't know whom you are working with."* —P9, Developer.

Teams are encouraged to keep photographs of all the members on a wall to get a 'team presence' that helps the members to recognise one another:

> *"It's very important to take pictures of [team member] and put them up on a card wall so that these people actually exists and become real in teams."* —P2, Agile Coach.

Teams also create online repositories or Wikis, where the photographs of the entire team and description of each member, are shared with the team. The Wiki gives a personal touch to the team members, and fosters more meaningful interaction:

> *"We have a "team place" or Wiki where we upload the photographs of the team members, and share some moments of [our] personal life which will help us move forward in our professional interaction."* —P24, Agile Coach.

Some teams allocate a short duration of time before daily meetings for team members to talk about personal matters, or to have some fun conversation:

> *"We need to have some personal time with other team members. We have 15 minutes before the daily meeting to speak freely to each other in the team. "* —P24, Agile Coach.

Teams exhibit strong dynamics when members are interacting without barriers. This kind of frequent interaction promotes team building:

> *"... the first 15 minutes [of daily meetings] was open time, and we could talk about anything we want. And, that's when we started seeing a very strong team building."* —P1, Developer.

Knowing members personally promotes better understanding in the team. Team members from all the locations are able to understand the difficulties faced by other members:

> *"Team members understand and value the other team member's life, had the rapport with all team members, [and] team members [were] well jelled with each other."* —P24, Agile Coach.

Teams need to participate in daily standup and retrospective meetings. The understanding established between team members allows them to 'share the pain' when working across different time zones:

> *"We start rotating the standup meeting [that is] for one month it is going to be at night [in India], and the next month it will be in the morning."* —P18, Business Analyst.

The interaction of the team improves significantly when team members know each other. The team should have seen all the members of the team, talked to them, and possibly worked with them in close proximity to develop strong team dynamics.

4.3   Open Communication

Participants encourage open communication in distributed teams — team members keep direct and honest communication within project team, and also with customers and management:

> *"Communication needs to be kept as open as possible, and there should not be any hierarchy so that [team members] can communicate directly with customers and management."* —P29, Scrum Master.

Open communication improves team interaction, and encourages team to be involved in decision making for the project:

"The project team believed in communicating very openly and transparently. So, all decisions were made in consultation with the entire project team." —P38, Scrum Master.

Participants realised that frequent open communication fosters good understanding between project team and management:

"The more we have open conversations, [the] better we understand the management, [and] then we are able to suggest better alternatives to them." —P17, Developer.

As a result of a good understanding between project team and management, participants were able to communicate directly with the management to make a request for the team:

"[When] I had to negotiate with the management, it was not a challenging thing [because] management well understood us. So it became easier for them to give [the request]." —P34, Agile Coach.

Some team members, however, face difficulties in engaging courageously in open communication with other members from different locations and the customers. Western participants described that their Indian counterparts unrealisticly agree to every request from the customers because it is typically not in the Indian culture to say 'No' to elders in a family, or superiors in an organisation:

"The Indians don't say 'No' to anything. That's one of the major problems faced by all the western customers. This is because of the culture [that] you should always obey the seniors." —P33, Agile Coach.

In order to address this concern, some teams engage in coaching to grow courage for team members to speak up, and improve interaction in the team:

"Here in India, trying to grow the courage for people to speak out and ask for what they need and be honest about what they can sustain, is something that I'm coaching a lot." —P31, Agile Coach.

Participants recognise that Agile methods value courage and open communication. Team members should be honest and transparent in all the levels of interactions, especially with customers:

"Most of the time, the members [in India] have a tendency to follow [requests] from the onsite members, but the onsite members are often more interested to know what other options are available. Agile taught them courage in speaking openly with the clients [sic]." —P24, Agile Coach.

Teams members should understand that courage is the foundation of open and honest communication — both within the project team, and with customers and management — and should strive to grow courage to facilitate communication.

4.4 Team Collocation

Agile methods prefer collocated teams to allow frequent interaction between the team members. While a distributed development team is not (by definition) collocated, many projects choose to collocate all the team members at the beginning of a project:

> "At the beginning of the [distributed] project, it's important for the entire team and the customer to be collocated for the [first] few weeks of the project. That's really important." —P20, Management Team.

Some teams collocate at the customer location for the first iteration to allow frequent interaction between the project team and customer:

> "The idea was to start all together as a whole team here [at customer location] for the first iteration in order to have direct interaction with customer." —P7, Agile Coach.

Collocating for the first iteration, or for a couple of weeks, helps the team to establish trust and build team relationships. When the members are sent back to their distributed locations, the trust and team relationships that have been developed during the collocation help them to interact effectively:

> "I would collocate a team for the first few weeks of the project [until] the team is able to build trust, build relationships, [and] build shared understanding. It is much easier to have conversations with team members on the phone if you've met them previously in person." —P22, Scrum Master.

There are some teams that rotate the location for team collocation between the customer location, and project team locations for a specified time duration:

> "We prefer to collocate. The first set of collocation involved the delivery team [from India]. Then the second time is the team from the USA, the customers and all the stakeholders came down here [to India] and worked from the same location for a month." —P18, Business Analyst.

Team collocation develops strong team relationships that increase team performance when team members get distributed in different locations:

> "When we started, we moved everyone to client site, [and] worked collocated. When we moved back to [our] site, there was a very natural bonding between the entire team, and we were doing an excellent job." —P35, Agile Coach.

Teams that are not able to collocate all the members for the first iteration would at the least send the senior members to initiate team interaction between members in all the locations:

> "Some of the senior members in the team go there [to other location] just to have a feel about the team members there." —P24, Agile Coach.

When team members travel to different locations, some team members were willing to spend personal time to get to know others and build strong team relationships:

> "*Apart from work, we spend a lot of good time with them. Some were keen to see our village life [in India] and come to our home.*" —P26, Agile Coach.

Some teams organise team building activities to accommodate the team members whom travel from other location. Team building activities encourage team members to interact comfortably, motivate them to develop good teamwork, and inspire them to work effectively as an Agile team.

> "*If other team members from onsite [are] coming here, we plan our team building activities so that we do that activities in that part of the month so that we can create a rapport with the onsite team.*" —P24, Agile Coach.

Realising the benefits of team collocation, some teams go so far as to move to the client's location for the entirety of short projects:

> "*We may not do the work offshore for the projects running for a smaller duration. We do it at the onsite [customer location] itself. We finish up the project from the client's site, and then come back [to our location].*" —P36, Quality Analyst.

Overall, participants found that team collocation, even for a short duration, facilitates team interaction that develops good teamwork and establishes trust across the whole team, supporting the 'one team' mindset that is crucial in a distributed Agile development project.

4.5 Team Ambassadors

Rather than collocating the whole team, individual team members can travel to the other team locations, to interact closely with the team members there. These team members, referred to as 'team ambassadors', travel solely to foster interpersonal relationships within the team. Team Ambassadors do not act as a managers or liasions between separate teams — the 'one team' mindset helps ensure team coordination and decision making is shared across the whole distributed team, primarily via the daily distributed meetings.

Participants describe that the main responsibility of the team ambassadors are to understand the team members in the location to which they have been sent:

> "*We wanted mainly to understand the team. When I went there, I started observing people and their way of interacting with each other. We started understanding each other, and started to work as a team.*" —P33, Agile Coach.

While working in the other location, the team ambassadors develop good relationships with the members there, promoting the dynamics of the entire team:

> "*When you send people over, you work with them, you go out with them, drink with them. In that way, you build this friendship, [and] you understand the people that you work with. So the interaction improves a lot.*" —P12, Developer.

Some teams rotate team ambassadors between the offshore and onsite locations:

> "*Developers will rotate with developers, and rotations happens between Business Analysts also. We have people from here rotating for some duration. And, this [rotation] promotes team dynamics here.*" —P16, Scrum Master.

and this rotation provides opportunities for more team members to act as ambassadors:

> "*We rotate [team ambassadors] to facilitate more conversations [with team] and be able to understand them better.*" —P37, Developer.

Participants P1, P3, P11-P20, P24, P27, P33, P36-P38 understand the importance of team ambassadors for distributed teams. The team ambassadors promote interaction, create rapport within the 'one team', helping members to work effectively together, even though they are distributed.

4.6 Coach Travels

The role of an Agile Coach, though self-descriptive, is to help a team or individual adopt and improve Agile methods and practices. A coach helps team members reflect and improve the activities involved in software development, and often withdraw from the team when the time is right and let the team continue:

> "*I coach the team who are adopting Agile. Often I guide others to deal with the situation at hand, but I want them to be in-charge of their own situation, and be independant of the coach.*" —P2, Agile Coach.

Coaches typically emphasize the importance of working together as 'one team', cultivating team spiring, and engaging team members to improve the team dynamics:

> "*[After] we had the coaching activities, we were successfully able to form the teams, and I could see good team dynamics happening. That [coaching] brought in a lot of changes within the team.*" —P33, Agile Coach.

Agile coaches travel around all the team's locations to meet all the team members and establish good relationships with them. The personal interaction and the bonding with team members allows the coach to engage in coaching activities even from remote locations:

> "*It is very difficult to coach someone [whom] you don't have any personal connection [with]. So, I think that going to other country and meeting the team members helps me to keep on coaching daily from here. You need to keep a team relationship and travel as much as possible if you are coaching.*" —P7, Agile Coach.

Coaches improve interaction amongst team members and develop team understanding in different locations:

"We had a coach from onsite who came here [to India] for several weeks. That coaching improved the interaction with remote team because the coach helped them to understand the working style of the remote team members. We prefer that the coach from remote location visits our team frequently." —P28, Management Team.

Coaches travelling allows them to appreciate the wider environment at each location. This allows the coach to have more informed conversations with customers or the rest of the team:

"Not only have I met more people in India, I actually know more about what India is like. And knowing that gives me better empathy and sympathy for the team that work under [difficult] conditions. " —P22, Agile Coach.

Coaching activities can also foster the dynamics between project team and support groups in the organisation:

"… make sure that interaction between development team and the support groups enable cross-communication. So, if something goes wrong, there are different groups within the same organisation to support [the team]." —P34, Agile Coach.

Participants P1, P2, P7, P22, P24, P28, P31, P33, P34, P35 and P39 acknowledge that the mentoring or coaching provided by Agile Coaches is vital to increase interactions within the project team, and with other groups in the organisation. The coaches need to travel to all the locations where a team is distributed so that personal connections can be maintained between coach and team members.

5   DISCUSSION AND RELATED WORK

Distributed teams should maintain as far as possible a single team identity across all locations to promote interaction and encourage cooperation amongst team members [38]. Loss of teamness' could pull distributed projects apart as it is often difficult to integrate separate independent teams into a coherent team. We found that the 'one team' mindset is the fundamental strategy that brings together the team members across different locations and encourages cooperation between the team and the customer.

Distributed teams require significant effort to promote team interaction between team members across time and space. Team members often leverage technology-mediated communication for team interaction. Non-verbal communication such as body language, hand gestures, facial expressions, and eye-contacts that forms 93% of communication are, however, missing in technology-mediated communication [39]. Fiore et al. [2] asserts that team interaction in distributed teams affects teamwork and team performances. Team building and establishing trust are difficult when team members do not work together in close proximity.

Although face-to-face interaction is still the best means of communication, it is expensive and difficult to schedule face-to-face meetings in distributed teams [6, 40]. Therefore, video-conferencing is often used as an alternative to face-to-face meetings to capture the visual aspects of communication such as hand gestures or facial expressions [40, 41]. While most teams used video-conferencing,

collocation, or team ambassadors to bind teams together, we found that some team members (P6, P8, P9, P25) have not even seen the faces of all members in the team. This is mainly because some team members did not get to travel to different locations to meet other members, and the technology-mediated communication between members in different locations were limited to phone calls or emails, but not video-conferencing. We found that teams need to create 'team presence' to allow the natural bonding between members in different locations. Practices such as keeping photographs of all the members on a wall, or maintaining Wikis with the photographs and description of the members create 'team presence' and build team ties.

Layman et al. [11] describe that a key member of the distributed Agile team who is physically located with the other team can provide an essential two-way communication conduit. This key member acted as a communication bridgehead between team members from different locations, and played the advocate for both groups on a daily basis. Braithwaite and Joyce [38] describe that local representatives travel from one location to another for an extended period to understand the members in that location, and share business domain knowledge between locations. We found that team ambassadors travel from one location to another, and work in close proximity with team members for a period of time. Unlike Layman et al., and Braithwaite and Joyce, these team ambassadors do not act as communication conduits, technical leads, Agile coaches or project managers. Rather, teams members communicate directly with each other, using video, audio, messaging, and email to contact remote team members, both in the daily stand-up meetings, and whenever other interactions are required. The ambadassors carry out their own development tasks, and strive to develop good team relationships and to promote direct interaction between local and remote team members.

6 LIMITATIONS

The inherent limitation of a Grounded Theory (GT) study is that the results are grounded in the specific contexts explored in the research [42, 43]. These contexts were dictated by the availability of the participants, and by our choice of research destinations. We do not claim that our results are universally generalisable to all distributed Agile software development projects, but rather our results accurately characterize the contexts studied.

7 CONCLUSION

We investigate the strategies for promoting team interaction in distributed Agile teams through a Grounded Theory study that involved 40 Agile practitioners from 24 software companies in the USA, Western Europe, Australia and India. Through rigorous analysis, we found distributed teams use six eminent strategies to promote interaction: *'one team' mindset*, *personal touch*, *open communication*, *team collocation*, *team ambassadors*, and *coach travels*. We are mindful that there can be other strategies to promote team interaction that can be useful and effective in their own contexts, but did not emerge from our analysis. The teams that we studied were found practicing at least one of these strategies to promote team interaction between members in different locations. Some teams proactively adopted these strategies to work effectively in distributed Agile projects, and some teams adopted them as solution strategies when problems around team interaction arise.

Future studies can compare and contrast team dynamics for distributed teams against team dynamics for collocated Agile teams.

References

[1] Castka, P., Bamber, C., Sharp, J., Belohoubek, P.: Factors affecting successful implementation of high performance teams. Team Performance Management **7**(7/8) (2001) 123–134

[2] Fiore, S.M.: Distributed coordination space: Toward a theory of distributed team process and performance. Theoretical Issues in Ergonomics Science **4**(3–4) (2003) 340–364

[3] Espinosa, J.A., Slaughter, S.A., Kraut, R.E., Herbsleb, J.D.: Familiarity, complexity, and team performance in geographically distributed software development. Organization Science (Providence, R.I.) **18**(4) (2007) 613–630

[4] Johnson, D.W., Johnson, F.P.: Joining Together: Group Theory and Group Skills. 4 edn. Prentice-Hall, Englewood Cliffs, NJ, US (1991)

[5] Katzenbach, J.R., Smith, D.K.: The Wisdom of Teams: Creating the High-performance Organization. Harvard Business School Press, Boston, Mass (1993)

[6] Korkala, M., Abrahamsson, P.: Communication in distributed Agile development: A case study. In: 33rd EUROMICRO Conference on Software Engineering and Advanced Applications. (2007) 203–210

[7] Herbsleb, J.D., Mockus, A.: An empirical study of speed and communication in globally distributed software development. IEEE Transactions on Software Engineering **29**(6) (2003) 481–494

[8] Korkala, M., Pikkarainen, 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

[9] Mockus, A., Herbsleb, J.D.: Challenges of global software development. In: Proceedings of the Seventh International Software Metrics Symposium. (2001) 182–184

[10] Prikladnicki, R., Audy, J.L.N., Damian, D., de Oliveira, T.C.: Distributed software development: Practices and challenges in different business strategies of offshoring and onshoring. International Conference on Global Software Engineering (2007) 262–274

[11] Layman, L., Williams, L., Damian, D., Bures, H.: Essential communication practices for Extreme Programming in a global software development team. Information and Software Technology **48**(9) (2006) 781–794 Special Issue Section: Distributed Software Development.

[12] Agerfalk, P.J., Fitzgerald, B.: Flexible and distributed software processes: Old petunias in new bowls? Communications of the ACM **49**(10) (2006) 26–34

[13] Moe, N.B., Dingsoyr, T., Dyba, T.: A teamwork model for understanding an Agile team: A case study of a Scrum project. Information and Software Technology **52**(5) (2010) 480–491

[14] Abbas, N., Gravell, A.M., Wills, G.B.: Historical roots of agile methods: Where did agile thinking come from? In: Agile Processes in Software Engineering and Extreme Programming. Volume 9 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2008) 94–103

[15] Beck, K.: Extreme Programming Explained: Embrace Change. 2 edn. Addison-Wesley, Upper Saddle River (2004)

[16] Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall PTR, Upper Saddle River, NJ, USA (2001)

[17] Highsmith,III, J.A.: Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House Publishing, New York, USA (2000)

[18] Stapleton, J.: Dynamic Systems Development Method: The Method in Practice. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1997)

[19] Coad, P., de Luca, J., Lefebvre, E.: Java Modeling Color with UML: Enterprise Components and Process with CDROM. Prentice Hall PTR, Upper Saddle River, NJ, USA (1999)

[20] Palmer, S., Felsing, M.: A Practical Guide to Feature-Driven Development. Pearson Education (2001)

[21] Cockburn, A.: Crystal Clear: A Human-Powered Methodology for Small Teams. Addison-Wesley Professional (2004)

[22] 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

[23] Fowler, M., Highsmith, J.: The Agile manifesto. Software Development **9**(8) (2001) 29–30

[24] Cockburn, A.: Agile Software Development. Addison-Wesley, Indianapolis (2002)

[25] Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methodologies. Communications of ACM **48** (May 2005) 72–78

[26] 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

[27] Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Sociology Press, Aldine, Chicago (1967)

[28] Paasivaara, M., Lassenius, C.: Could global software development benefit from Agile methods? In: Proceedings of the IEEE International Conference on Global Software Engineering, Washington, DC, USA, IEEE Computer Society (2006) 109–113

[29] 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

[30] Martin, A., Biddle, R., Noble, J.: The XP customer team: A grounded theory. In: Proceedings of the AGILE. (2009) 57–64

[31] Whitworth, E., Biddle, R.: The social nature of Agile teams. In: Proceedings of the AGILE, Washington, DC, USA, IEEE Computer Society (2007) 26–36

[32] Ramesh, B., Cao, L., Mohan, K., Xu, P.: Can distributed software development be Agile? Communication of the ACM **49**(10) (2006) 41–46

[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. AGILE '08. Conference. (2008) 63–72

[34] Allan, G.: A critique of using grounded theory as a research method. Electronic Journal of Business Research Methods **2**(1) (2003) 1–10

[35] Georgieva, S., Allan, G.: Best practices in project management through a grounded theory lens. Electronic Journal of Business Research Methods **6**(1) (2008) 43–52

[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] Braithwaite, K., Joyce, T.: XP expanded: Distributed extreme programming. In Baumeister, H., Marchesi, M., Holcombe, M., eds.: Extreme Programming and Agile Processes in Software Engineering. Volume 3556 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2005) 1524–1526

[39] Bianchi-berthouze, N., Kleinsmith, A.: A categorical approach to affective gesture recognition. Connection Science **15**(4) (2003) 259–269

[40] Dorairaj, S., Noble, J., Malik, P.: Understanding the importance of trust in distributed Agile projects: A practical perspective. In: Agile Processes in Software Engineering and Extreme Programming. Volume 48 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, Trondheim, Norway (2010) 172–177

[41] 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

[42] 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

[43] Hoda, R., Noble, J., Marshall, S.: Developing a grounded theory to explain the practices of self-organizing Agile teams. Empirical Software Engineering (2011) 1–31