Exploring communication challenges associated with Agile practices in a globally distributed environment.

Sunila Modi, Pamela Abbott, Steve Counsell
Department of IS and Computing, Brunel University, Middx, UK.
Email: sunila.modi@brunel.ac.uk

Industrial Collaborator: Ben Stopford, benjamin.stopford@rbs.com

Abstract
Co-located agile teams rely heavily on physical proximity, face-to-face communication, coordination and close collaboration. These tenets of agile methods are hampered in globally distributed environments and the shortcomings have to be consciously addressed and supported through various activities between the onshore and offshore locations. This paper reports on preliminary findings of the communication processes in one case study which highlights the necessity of providing multiple modes of communication to support the shortfalls of physical proximity and curtailed face-to-face interactions in distributed agile teams.

1.0 Introduction
In the last decade, the number of organisations investing heavily in global markets as part of strategic planning has significantly increased (Herbsleb, 2007). The move to transform national markets to global markets is growing at a tremendous rate. Organisations are therefore trying to reducing software development costs by distributing projects in offshore locations in order to gain competitive advantages; the other main drivers are greater productivity and flexibility. While offshoring and outsourcing is becoming the norm in many organisations, global software teams still face significant challenges at various levels responding to dynamic business environments and evolving user requirements. Hence, global software development teams are continually pursuing methods which provide flexibility to software methods yet provide enough rigour compared to traditional waterfall approaches (Agerfalk & Fitzgerald, 2006).

Since the Agile Manifesto was published over a decade ago (Fowler & Highsmith, 2001), there has been a gradual stream of literature on the use of Agile methods in distributed settings illustrating the challenges in this field. Agile software practices are based on developing continuous working software in an iterative manner, accepting user requirements as user stories at any point during project development and removing the inflexibility associated with traditional waterfall model. Such practices intend to provide greater flexibility by avoiding up-front design; they are therefore they could be advantageous in a changing business environment.

One of the main themes in offshoring literature is the need for further research in distributed agile environments in practice to better understand such practices (Dybå & Dingsøyr, 2008; Abrahamsson et al., 2009; Agerfalk et al., 2009; Dingsøyr et al., 2012). This paper reports on a case study of a globally distributed agile team which are collaborating together. The study provides valuable insights into how agile methods are applied and adopted and the on-going challenges that distributed teams face. The research focuses on the following questions:

- How do global agile teams coordinate, adapt and adjust agile practices in a globally distributed environment?
• What ongoing communication challenges do teams face in the application of agile methods in a given context?

The remainder of this paper is organised as follows: Section 2 gives an overview of related research on agile methods in globally distributed environments, the challenges it presents and support mechanisms available to adapt agile methods. Section 3 describes the research method and approach used in this study. Section 4 presents the findings so far followed by a discussion in Section 5 highlighting key insights of how agile methods are adapted in practice. Section 6 concludes by outlining further work.

2.0 Related Research

2.1 Agile methods
Agile methods arose from the need to address some of the basic problems of traditional software development, specifically the lengthy time period between inception and implementation of traditional methods. Typically, actual project costs vary significantly from estimated project costs; deliverables rarely meet the user expectations because requirements change over the course of the project. One of the key reasons why these problems arise is that traditional plan-driven methods adopt a sequential approach to software design and development (Agerfalk & Fitzgerald, 2006), whereas with agile methods the emphasis is based on its core values and principles.

The core values of the Agile Manifesto (Fowler & Highsmith, 2001) are:

• “Individuals and interactions over processes and tools.
• Working software over comprehensive documentation
• Customer collaboration over contract negotiation.
• Responding to change over following plan.”

These values are fundamental and govern the agile software development process. They are popular because they provide flexibility and establish close communication with users. They minimise risk since working software is delivered in increments and therefore priorities can be re-evaluated at the end of each cycle. Continuous code integration allows feedback on consistent testing and errors are thus eliminated far earlier in the project lifecycle. The modular nature lends itself to object-oriented designs and tasks can be mutually shared between teams (Ambler, 2002). While there are reasons to support both methods –traditional methods take a predictability stance and Agile methods take a flexibility standpoint with “just enough” rigour.

There are a number of approaches which come under the group of Agile methods. Scrum (Schwaber & Beedle, 2002) and Extreme Programming (XP) (Beck, 1999) are most widely used in distributed settings. XP has been described as a “light-weight” methodology for small-to-medium-sized teams where software is developed in the face of vague or changing requirements. XP consists of twelve practices, namely Planning Game, Small Releases, Metaphor, Simple Design, Testing, Refactoring, Pair Programming, Collective Ownership, Continuous Integration, 40-hour Week, On-Site Customer, and Coding Standards. Test Driven Development (TDD) is usually combined with XP since it advocates writing test cases before the code is developed in facilitating improved design and refactoring (Beck & Andres, 2004).
2.2 Challenges faced by distributed agile teams.

A variety of literature on offshoring focuses on challenges in a global software development environment. These challenges originate because of an increase in temporal, geographical and socio-cultural distance. This hampers communication, coordination and the control processes of software projects (Carmel & Agarwal, 2001; Agerfalk & Fitzgarald, 2006). Table 1 gives a summary of identified challenges.

<table>
<thead>
<tr>
<th>Processes</th>
<th>Identified Challenges</th>
<th>Caused by (Distance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>Reduced opportunity for synchronous communication</td>
<td>Temporal</td>
</tr>
<tr>
<td></td>
<td>Face-to-face meetings difficult</td>
<td>Geographical</td>
</tr>
<tr>
<td></td>
<td>Greater risk of cultural misunderstandings</td>
<td>Socio-cultural</td>
</tr>
<tr>
<td>Coordination</td>
<td>Increased coordination costs</td>
<td>Temporal</td>
</tr>
<tr>
<td></td>
<td>Reduced informal contact can lead to lack of critical task awareness</td>
<td>Geographical</td>
</tr>
<tr>
<td></td>
<td>Inconsistent work practices can impinge on effective coordination</td>
<td>Socio-cultural</td>
</tr>
<tr>
<td></td>
<td>Reduced cooperation arising from misunderstanding</td>
<td>Socio-cultural</td>
</tr>
<tr>
<td>Control</td>
<td>Managing project artefacts may be subject to delays</td>
<td>Temporal</td>
</tr>
<tr>
<td></td>
<td>Difficult to convey vision and strategy</td>
<td>Geographical</td>
</tr>
<tr>
<td></td>
<td>Perceived threat from training low-cost rivals</td>
<td>Geographical</td>
</tr>
<tr>
<td></td>
<td>Different perceptions of authority can undermine morale</td>
<td>Socio-cultural</td>
</tr>
<tr>
<td></td>
<td>Managers must adapt to local regulations</td>
<td>Socio-cultural</td>
</tr>
</tbody>
</table>

Table 1: Identified Challenges in Globally Distributed Software Development
(Adapted from Agerfalk & Fitzgerald 2006)

Application of Agile methods in a distributed environment is even further challenged since it conflicts with the need for physical proximity and reliance on face-to-face communication an essential precept of agile methods (Fowler & Highsmith, 2001; Beck 1999; Layman et al., 2006). Further challenges can also arise in some of the Agile practices such as pair programming where in co-located teams two members work side by side on the same code; remote pairing could prove to be more difficult due to difference in time zones and differing technical backgrounds (Yap 2005; Dhoke et al., 2011). To integrate Agile methods into global distributed setting they have to be adapted and tailored to the needs of the individual contextual setting (Kircher et al., 2001; Fitzgerald, 2006; Layman et al., 2006; Lee et al., 2006; Ramesh et al., 2009; Batra, 2009; Dhoke et al., 2011). The following section outlines some of suggestions made in the literature.

2.3 Practices supporting distributed agile methods

Frequent Visits & Staff Rotation

To build and maintain good collaborative relationships and reduce cultural misunderstandings frequent visits have to be made to the offshore teams (Fowler, 2006; Ramesh et al., 2006; Miller, 2009). Fowler (2006) describes two types of visits: intense seeding visits carried out early in the project and maintaining visits which are shorter and have the aim of continuing and maintaining relationships. It is difficult to build up trust without physically meeting people and therefore such visits foster trust and team cohesion (Yap 2005; Ramesh et al., 2006). Other researchers have also suggested the importance of rotating team members between locations (Braithwaite & Joyce, 2005; Danait, 2005). Thoughtworks reported that having “face-time” and “staff rotation” was considered best practice to gel and create team cohesion (West et al., 2011).
Building a shared understanding

One of the twelve practices of XP is to build a notion of shared understanding within the team; pair-programming can contribute to this practice (Beck & Andres, 2004). In co-located teams this is easily facilitated as the interactions between developers happen in the same space and time which allows for mental models and skills to be adjusted to common understandings. Although, studies have revealed that a number of the factors come into play such as developers’ experience, system complexity, interruptions and time pressure (Arisholm et al., 2007; Plonka et al., 2012). This process is further challenged in distributed environments as social ties have to be fostered and nurtured over space and time barriers for successful collaborations (Kotlarsky & Oshri, 2005). Flor (2006) advise having various communication channels within the pairs can reduce the proximity of space. Bass et al. (2007) suggest having a “collaboration maturity” within a team where relationships have had time to develop. Sutherland (2008) suggested the “One Team” mind-set working across different locations to help lessen the barriers of “them” and “us”.

Providing Multiple Communication Modes

To alleviate the problems associated with geographical distance between distributed teams a number researchers have proposed multiple communication modes for enhancing communication channels (Kircher et al., 2001; Braithwaite & Joyce, 2005; Ramesh et al., 2006; Layman et al., 2006; Sutherland 2007; Abbattista et al., 2008; Prikladnicki et al., 2012). Prikladnicki et al. (2012) categorise supporting software technologies as general purpose tools such as email, instant messaging (IM) systems, video conferencing tools and screen sharing tools. Secondly, Web 2.0 tools such wikis, blogs and virtual white boards which can aid in knowledge sharing. A final category comprises of software engineering and project management tools including continuous integration systems, build systems, issue tracking tools and project management systems. Such tools support the control and collaboration aspects of projects, giving team members (regardless of location) a real-time progress of the project iteration by showing the progress of current user stories. It also tracks different tasks and an outline of errors which need to be resolved. As a result, creating shared “places” which raises awareness of collaboration and effective structures. This can help teams to coordinate, collaborate and generate suitable workflows which then become the “information radiators” for distributed teams (Cockburn, 2002; Berczuk, 2007). Sharp & Robinson (2008) discuss the importance of the physical artefacts, notably the “wall” and “cards” in co-located teams and how they tie in with the core philosophy of XP of shared responsibilities and collective ownership. In distributed teams, shared places and a “virtual wall” have to be created where cards can be seen by all the members of the teams in different locations (Berczuk, 2007).

Team Composition

Having team members with prior experiences of agile practices is a valuable asset since it promotes similar mind-sets and fosters Agile values (Yap, 2005; Sutherland, 2008). Some studies have suggested that having a similar composition at both locations can reduce dependency on one location (Hogan, 2006; Sutherland, 2008). These practices imply that at each location there is an equivalent person who would have similar understandings about their roles. West et al. (2011) state the importance of establishing the right team composition and recommend two Business Analysts (one at each location) who can communicate on a regular basis.

Synchronising work patterns

The temporal distance can also cause lack of overlapping working hours, which in turn can cause problems for lack of synchronous communication and future misunderstandings.
Suggestions have been made to synchronise work patterns to maximise the overlap and increase the communication bandwidth (Ramesh et al., 2006; Paasivaara et al., 2009). From a strategic point of view, strong management commitment has to exist to address the challenges of agile methods in globally distributed environments.

3.0 Research Approach
The objectives of this research are twofold; firstly to gain a deeper understanding of adopted Agile methods for a distributed development project in practice and, secondly, to identify the ongoing communication challenges of agile methods in distributed environments. An interpretative approach was adopted for the study since it best suits investigating the application of agile methods in practice. Such an approach is suitable where there is a complex relationship between people and processes and where researchers are interested in the subjective understanding of the participants (Walsham, 2006). A case study approach allows an authentic representation of the situation to be explored (Benbasat et al., 1987; Yin, 2003). Case studies permit “thick descriptions” (Yin, 2003) which provide rich and valuable findings.

3.1 Contextual Setting
The case-study is based on an international bank and financial services company anonymised to ABC Bank. Its headquarters are in the UK and it has distributed business centres and IT centres across the globe. Strategically, the bank has set up captive centres in India where a number of teams work on an offshore basis; all employees belong to one organisation. The case study was selected for investigation because it uses agile methods in distributed settings. In total, the team consists of twenty developers across the two sites (London and India). The London team consists of the Project Manager, two Business Analysts, Technical Lead, six Senior Software Developers and two QA staff. The India team is considerably larger, consisting of a Business Analyst, a Technical Lead, a number of junior and senior developers and the majority of the QA team. The project team is developing a central depository system called the Operational Data Cache (ODC) system for all the different types of trades carried out by the bank. The research on this project began approximately two and half years after its inception.

3.2 Data collection
The data collection involved a sixteen of semi-structured interviews which lasted, on average, one hour each. Eight face-to-face interviews with individuals from the London team and eight video-conference interviews with the India team counterparts. The interview protocol served as an instrument to enable consistency in the data collection and enhance reliability of the results. The interviews focused on the following areas:

- Sharing Processes and practices relating to team coordination and collaboration in both locations: relating to communication;
- Knowledge within both teams and the continuing challenges the team was facing.

The interview questions were formulated beforehand; however a large degree of flexibility was incorporated to allow the researcher to pursue relevant issues that arose during the interview and to allow open-ended answers from the interviewees. The interviews were recorded and transcribed fully later.

3.3 Data Analysis
Data analysis was carried out using Template Analysis since it takes a “contextual constructivist” stance and works well when comparing different perspectives within a specific context (King, 2004). The first step involved reading and re-reading all the transcripts. An
initial template was then constructed creating high level codes which were then sub-divided into lower order codes. The initial template was used and revised to insert/modify codes. QSR-NVIVO software was used to organise and examine the data in detail.

4.0 Findings
4.1 Adapting XP practices within the team
The ODC team within ABC Bank, adopted the tenets of XP with TDD from the start of the project that provided them with a basis of structure in defining and adapting the agile processes. The essential practices of XP of pair programming, iterative development, little up-front design, unit testing and continuous integration were applied with varying degrees. Some of team members in London have a similar IT background having previously worked at Thoughtworks and this provided them with a firm foundation of working with agile approaches in distributed environments. Other team members had used agile practices in previous projects and had similar capabilities. Initially, the India team was less familiar with agile practices and they were mentored and exposed to XP practices by the London team.

Daily Stand ups
Each location has its own separate daily stand up meetings due to the time differences; although Business Analysts and Technical Leads from London and India use teleconference calls to provide updates on a daily basis. These meetings discuss the daily progress and highlight any problems which can cause delays in development work and as a useful way of exchanging information within a team.

Pair Programming
At the start of the project, abridged pair-programing sessions were used as vehicle to develop and enhance the programming skills and to share knowledge about OO practices. It was carried out with two developers one from each location pairing on one user story for one hour per day. These sessions were facilitated by screen sharing devices and the use of the telephone. However, this process was hampered by technology - the screen sharing software did not allow for concurrent editing and this practice was perceived as frustrating for the developers. One of the interviewees stated …

… there was time when we did kind of pair-programming between London and India, but it was really hard – mainly because of the technology. It is really hard to share screens and when one person loads the screen and the phone line was pretty weak it was hard to hear. Even the language barrier; I am not British - I have got an accent so someone from India could not understand me and the other way round.

4.2 Supporting adapted XP practices.
To support the distributed teams the following practices were put in place.

Frequent visits and staff rotation
Exchange visits were organised for developers to be part the offshore team for two weeks, to build a rapport, develop relationships with the team members with the offshore location and contribute knowledge sharing sessions. This also provided the Indian team members with an opportunity to meet some of London team members in person and gain a deeper understanding of the domain knowledge. However, at the time of the study no further visits could take place as there was a freeze on travelling expenses within the organisation.

Building a shared understanding
When the project first started, the offshore team was less familiar with Agile approaches and the London team mentored and supported them, for them to gain a better shared
understanding of the project. This was accomplished by using Dreyfus’s knowledge acquisition model (Dreyfus & Dreyfus; 1986) which consists of five developmental stages: Novice, Advanced Beginner, Competent, Proficient and Expert. The different stages reflect changes in aspects of performance from abstract to concrete experiences and from the detached to involved performer. As the team was going through the initial stages, there were misunderstandings which caused frustration and friction between developers. However, this has gradually improved over time and knowledge sharing has taken place though the local-global interfaces (Abbott et al., 2012; Sahay et al., 2003) and developer exchange visits subdued some of the barriers. One interviewee mentioned …

when we started there was a huge barrier, just initially from not knowing who’s who. You’re not quite sure who has done what code and you see some good bits and then you see some bad bits and then if you’re not careful you taint everybody with the bad pieces of code and you assume everybody’s like that. And that’s not always the case. So certainly going to India definitely helped.

At the start of project there was a definite intention of one team in two locations, rather than two separate teams. It was only after the exchange visits that that team felt more of attached and cohesive bonds were formed through spending time at the other location. One interview stated …

Before the exchange – it felt like two teams – it felt detached. After the exchange we met the people. It feels more like one team but obviously separated by time difference, distance and space. It feels much better now – it is very important obviously.

Team Coordination and sharing artefacts

To support the team coordination and collaboration the project teams use an issue tracking and project management system called JIRA (by Altassian) support the control and collaboration aspects of projects. It gives team members (regardless of location) a real-time visibility of the project iteration it documents the progress of current user stories for each iteration. A story is essentially an outline of one of the requirements for that iteration. It also offers management reporting tools. This creates a shared platform for both locations and a “virtual wall” which can help in the coordination and collaboration activity to generate appropriate workflows. It also allows Business Analysts and Project Managers to reprioritise tasks. One of the interviewees stated…

Our project is adopting the ‘Agile way of working’, what we are trying to do is move away from lots of up-front analysis… we are trying to get away from that because that is fairly unresponsive to change, [as] we do have a fair amount of change, not only in the detail of requirements but also in the priorities and … our whole work profile is based on user stories. We are recording those and communicating them in a system called JIRA… also you can add to these conversations. So our method tends to have a lot of checkpoints in it, where we force communication..[to be documented].

Table 2 summarises the practices carried out by the ODC team in relation to the suggested practices given in the literature for distributed agile teams. Column 3 highlights the challenges the supporting practice is trying to address and column 4 identifies the continuing challenges for the ODC team.
<table>
<thead>
<tr>
<th>Practices suggested for distributed Agile environment</th>
<th>ODC team adopted Agile Practice</th>
<th>Challenges this addresses</th>
<th>Continuing challenges within the ODC team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequent Visits &amp; staff rotation</td>
<td>Yes: taken up at the start of the project</td>
<td>Reduced opportunity for face-to-face meetings. Difficult to convey vision &amp; strategy. Greater risk of cultural misunderstandings. Different perceptions of authority can undermine morale.</td>
<td>Currently these visits have stopped due to budget cuts.</td>
</tr>
<tr>
<td>Building a shared understanding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Shared Daily stand-ups</strong></td>
<td>Yes: on a daily basis</td>
<td>Increased coordination costs. Reduced informal contact can lead to lack of critical task awareness. Reduced cooperation arising from misunderstanding.</td>
<td></td>
</tr>
<tr>
<td><strong>Pair programming across locations</strong></td>
<td>Yes: regularly at the start of the project</td>
<td>Inconsistent work practices can impinge on effective coordination. Reduced cooperation arising from misunderstanding.</td>
<td>Pair-programming across locations does not take place regularly now.</td>
</tr>
<tr>
<td><strong>One-team mind set</strong></td>
<td>Yes: the team identifies themselves as one team in two locations.</td>
<td>Greater risk of cultural misunderstandings. Difficult to convey vision and strategy. Reduced cooperation arising from misunderstanding.</td>
<td></td>
</tr>
<tr>
<td>Providing multiple communication modes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Email, instant messaging (IM) systems, video conferencing tools and screen sharing tools.</strong></td>
<td>Yes: the team uses email, IM systems on daily basis. The screen sharing tools are used when required.</td>
<td>Reduced opportunity for synchronous communication. Managing of project artefacts may be subject to delays. Reduced informal contact can lead to lack of critical task awareness.</td>
<td>Screen sharing software tools can cause frustration for the developers. Video conferencing is used infrequently.</td>
</tr>
<tr>
<td><strong>Web2.0 tools – wikis, blogs and virtual whiteboards.</strong></td>
<td>Yes – the team use a wiki to store essential documentation</td>
<td>Inconsistent work practices can impinge on effective coordination</td>
<td></td>
</tr>
<tr>
<td><strong>Issue and defect-tracking systems</strong></td>
<td>Yes – the use of JIRA</td>
<td>Managing of project artefacts may be subject to delays.</td>
<td></td>
</tr>
<tr>
<td>Team Composition</td>
<td>No - Majority of the senior team based in London.</td>
<td>Different perceptions of authority can undermine morale</td>
<td>Stakeholders and users are based in London.</td>
</tr>
</tbody>
</table>

Table 2 Summary of suggested practices for distributed Agile development and ODC team practices.
Ongoing challenges

A number of practices suggested for distributed agile development in literature have been adopted by the ODC team (Table 2), however a small number challenges continue to exist as the visits have stopped due to budget cuts, this can increase the socio-cultural separation and reduce the opportunities for face-to-face discussions. Face-to-face interactions are an essential tenet of agile practices (Highsmith & Cockburn, 2001) and if these are missing it may cause misunderstandings especially if the team members have not met all their counterparts in the other location (Paasivaara & Lassenius, 2006).

Secondly, the practice of pair-programming sessions across boundaries does not work well and causes frustrations among the developers owing to limitations of the screen sharing software and sometimes due to capability differences between developers (Batra, 2009). For distributed pair-programming to work effectively it is necessary to provide remote collaboration tools which can emulate a similar infrastructure to when programmers work in close proximity (Flor, 2006).

The ODC team use video-conferencing tools infrequently as in the past technical glitches have caused a number of problems. This again reduces richer face-to-face discussions which in turn can reduce the opportunity for speculative questions to be asked enabling sense-making and deeper understanding of the problems and it can also reduce the visibility of what people are doing at both locations (Prikladnicki et al., 2012; Abbattista et al., 2008).

Finally, the team composition is not the same in both locations as the stakeholders are based in London and majority of the senior team members are based in London. Therefore the Indian team members sometimes feel they have to wait for direction and decisions to be made by the London team (Levina & Vaast, 2008).

5.0 Discussion & Conclusion

The ODC project has been running for approximately two and half years. The overall experience of the team has been positive and the teams are motivated. They have matured through time once the exchanges visits took place and have since evolved as “one” team culture. However, there have been a number of hurdles that both teams have had to overcome during the course of time some of which were discussed earlier.

Some team members made suggestions of how the practice can be improved to increase visual presence of the offshore team, such as web cameras so that they can see the person they are communicating with, use of video walls which would give them the presence of the offshore team; using virtual Kanban boards or virtual whiteboards might be a positive step to visualise workflows. These suggestions emphasises the importance of visual cues of physical artefacts which are part of the of collocated XP teams (Sharp & Robinson, 2008).

At the start of the paper two research interview questions were raised: how do global agile teams coordinate, adapt and adjust agile practices in a globally distributed environment and what ongoing communication challenges do teams face in the application of agile methods in a given context. The findings from the case study suggest that agile practices have to be tailored and adapted to the context of particular distributed environments. It emphasises the necessity of providing multiple modes of communication to support the shortfalls of physical proximity and curtailed face-to-face interactions in distributed teams. Having team members who are adaptable to working in distributed settings in collaborative manner when needed to in order to accomplish team tasks. Furthermore, it draws attention to augmenting supporting practices to be continued (such as the frequent visits) throughout the project as such practices
underpin and reinforce the importance of building and maintaining shared understandings which in essence is one of the core values of the Agile Manifesto.

While one case study provides some insight to a better understanding of the communication processes in a distributed agile setting, it is however a limitation of this paper. Thus, further work needs to be carried out to enhance the understanding of how communication in multiple modes supports the essential tenets of coordination and collaboration in distributed agile practices.

**Acknowledgements**
The authors would like to thank the industrial collaborator for their support and the participants for their time.

**References**


