

Spring 3-19-2013

Raising Awareness In Distributed Agile Development - A Case Study Perspective

Sunila Modi

Brunel University, sunila.modi@brunel.ac.uk

Pamela Abbott

Brunel University, pamela.abbott@brunel.ac.uk

Steve Counsell

Brunel University, steve.counsell@brunel.ac.uk

Follow this and additional works at: <http://aisel.aisnet.org/ukais2013>

Recommended Citation

Modi, Sunila; Abbott, Pamela; and Counsell, Steve, "Raising Awareness In Distributed Agile Development - A Case Study Perspective" (2013). *UK Academy for Information Systems Conference Proceedings 2013*. 26.
<http://aisel.aisnet.org/ukais2013/26>

This material is brought to you by the UK Academy for Information Systems at AIS Electronic Library (AISEL). It has been accepted for inclusion in UK Academy for Information Systems Conference Proceedings 2013 by an authorized administrator of AIS Electronic Library (AISEL). For more information, please contact elibrary@aisnet.org.

RAISING AWARENESS IN DISTRIBUTED AGILE DEVELOPMENT - A CASE STUDY PERSPECTIVE

Sunila Modi

Department of Information Systems and Computing, Brunel University, London, UK
Email: sunila.modi@brunel.ac.uk

Pamela Abbott

Department of Information Systems and Computing, Brunel University, London, UK
Email: pamela.abbott@brunel.ac.uk

Steve Counsell

Department of Information Systems and Computing, Brunel University, London, UK
Email: steve.counsell@brunel.ac.uk

Ben Stopford

Industrial Collaborator

Abstract

Raising and maintaining awareness in distributed agile cross-cultural teams is a challenging process, especially as the tenets of agile methods rely heavily on physical proximity, face-to-face communication, coordination and close collaboration. This paper reports on preliminary findings on how a distributed agile team within the financial sector, engages in raising awareness to carry out collaborative activities to accomplish project tasks. We have adapted the use of 3C Collaboration model as an evaluative mechanism to examine how the different dimensions (communication, coordination and cooperation) of the model stimulate awareness within a distributed agile team. The insights gained from the case study, suggest there is a constant interplay between the offshore and onshore teams to try and raise and maintain awareness in order to achieve project goals.

Keywords: Distributed agile development, awareness, 3C Collaboration model, collaborative practices, communication and coordination.

1.0 Introduction

Global software development (GSD) and distributed software development (DSD) environments have become the norm for many global organisations to leverage skilled resources around the world, increase productivity, offer flexibility and reduce costs by operating in global labour markets (Carmel & Tija, 2005, Herbsleb, 2007). Despite the number of benefits this can offer, it also presents a number of challenges in terms of coordinating software projects across global sites in different time zones; software development teams have to deal with temporal, geographical and socio-cultural distance (Carmen & Agarwal, 2001; Agerfalk & Fitzgerald, 2006). These challenges can be accentuated by having to respond to dynamic business environments and evolving user requirements. Therefore 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). To achieve these goals, a number of organisations have started to use agile methods in distributed environments as a viable alternative to the waterfall approach. However, deploying agile methods in distributed environments can raise additional conflicts, as they were originally developed for collocated environments and the tenets for agile methods are: need for physical proximity, reliance on face-to-face communication, cooperation on all parts of the project and a suitable collaborative environment where trust and shared understandings can be developed through a team culture (Fowler & Highsmith, 2001).

1.1 Motivation

One key challenge that globally distributed cross-cultural agile teams face is developing an awareness of distributed team members and project task awareness. Awareness is defined as “*an understanding of the activities of others, which provides a context for your own activities*” (Dourish & Bellotti 1992). Having appropriate awareness in software teams of project activities, tasks, artefacts and expertise of team members within a project is essential since agile methods endorse collaborative and cooperative activities. In collocated teams, awareness is demonstrated by (a) team meetings (b) watching other developers carry out their tasks (c) observing changes made to project artefacts (d) effective communication, coordinating and collaborating with colleagues to complete project tasks. Ko et al., (2007) reported that in collocated development environments most frequently sought after information included

awareness about artefacts and co-workers, and the developers rely on co-workers as their most frequent information sources. Whereas, in distributed team members can face a lack of team and task awareness and generally experience more difficulties in establishing and maintaining awareness needed for the progress of tasks, the artefacts and their respective offshore team members (Herbsleb, 2007; Damian et al. 2007).

The motivation of this study is to seek a deeper understanding of how awareness is raised within distributed agile teams and how it is maintained in practice. The study has been carried out using a case-study approach based on an international bank, which we will call ABC Bank (the name is anonymised for confidentiality purposes) 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 the team members are employed by the bank. This paper focuses on the investigating following research questions:

1. *How do distributed cross-cultural agile teams raise and maintain increased awareness for collaborative activities to take place across the boundaries?*
2. *What are the on-going challenges that distributed agile teams face to sustain awareness?*

The rest of this paper is structured as follows. Section 2 gives a brief review of agile values and principles and how they have to be adapted for distributed settings. We then discuss related work on awareness and introduce how the 3C collaboration model could be used as an evaluate instrument to explore how awareness is being fostered within distributed settings. Section 3 describes the research approach used for this study and its findings are presented in Section 4. This is followed by a discussion of the key insights in Section 5 and a concluding section drawing together the research contributions, limitations of the study and further work to be carried out.

2.0 Agile Values and Principles

The Agile Manifesto was published over a decade ago with core values of (Fowler & Highsmith, 2001):

- *“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 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 priorities can be re-evaluated at the end of each cycle. Continuous code integration allows feedback on continuous testing and errors are thus eliminated far earlier in the project lifecycle. The modular nature of Agile 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 while agile methods take a flexibility standpoint with “just enough” rigour. As agile methods provide this flexibility, they are more suited to dynamic business environments where the user needs and requirements are constantly changing and evolving. Underpinning the Agile Manifesto are twelve associated principles (Fowler & Highsmith, 2001) which can be seen as good practice and guidelines for agile teams. The practices place emphasis on changing to certain work environments within the teams (see Table 1).

Agile Principles	
P1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
P2	Welcome changing requirements, even late in development.
P3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to shorter timescale.
P4	Business people and developers must work together daily throughout the project.
P5	Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done
P6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversations.
P7	Working software is the primary measure of progress.
P8	Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.
P9	Continuous attention to technical excellence and good design enhances agility
P10	Simplicity - the art of maximizing the amount of work not done-is essential.
P11	The best architectures, requirements and designs emerge from self-organizing teams.
P12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Table 1: Agile Principles (Fowler & Highsmith, 2001)

Examining three of these principles (P4, P5 and P6) in more detail, entails having a raised awareness of the project tasks which in turn leads to close collaboration among the members of software team. P4 implies frequent communication, coordination and interaction has to take place within the development team and with users to deliver working software. P5 stresses the importance of providing agile teams with a suitable environment which allows teams to build on trust and develop a shared team culture. P6 highlights the need for collocated teams and face-to-face conversations within the software team.

2.1 Practices in Distributed Agile Development

In distributed agile teams, the Agile principles: P4, P5 and P6 (see Table 1), become much more difficult to accomplish as face-to-face communication and coordination cannot happen on daily basis, which can lead to a lack of awareness. Moreover, software teams in different locations share a reduced amount of contextual information and tend to have less information about what their team counterparts at the other location are doing. This in turn can lead to communication misunderstandings or breakdowns within the teams and impede progress on the project (Hersleb, 2007; Ramesh et al., 2006; Damian et al., 2007). Therefore in distributed agile settings, team members have to value the agile principles in theory but adapt them in practice (Nerur et al., 2005; Ramesh et al. 2006; Batra, 2009). A related stream of research has focused on tailoring and adapting agile methods to contextual needs of the organisation within collocated settings (Fitzgerald et al.; 2006; Cao et al. 2009).

2.2 Related research on awareness

Research on awareness has also been carried out extensively in the Computer Supported Collaborative Work (CSCW) field and several studies have suggested potential requirements, design and development of specific awareness systems which could be used in the GSD context (Sarma et al., 2003; Froehlich & Dourish., 2004; Biehl et al., 2007; Dullemond et al., 2009). Gutwin and Greenberg (2002) developed a framework to examine workspace awareness which is defined it as “*up-to-moment understanding of another person’s interactions with the shared workspace*”. This has been applied in distributed development literature, where all the elements of awareness are supported by one specific awareness system or as an assessment tool where a number of different awareness systems are being evaluated for potential users (Storey et al., 2005). Omoronyia et al., (2010) applied this framework to carry out a review of the current awareness systems for the global software engineering arena. However, the framework was designed for use in real-time, synchronous distributed groupware systems.

An alternative model of assessing and evaluating awareness is using the 3C Collaboration model (see Figure1), originally proposed by Ellis et al., (1991) and extended by Fuks et al., (2008). The model defines collaboration as “*the union of communication, coordination and cooperation efforts*”. The model is iterative in nature and although the three aspects are separated, there is continuous interplay between them. Steninmacher et al., (2010) applied the 3C Collaboration model to carry out a review within GSD, based on the three dimensions (coordination,

communication and cooperation). The review categorised a number of studies into one of the three dimensions. For the GSD context, they adopted the following meanings to the inter-related dimensions of awareness:

- **Communication** as “*the way the messages and information are exchanged among people, reducing any gaps, ambiguity or the effort needed to understand, establish or continue conversation*”
- **Coordination** as “*the support offered to people managing themselves or being aware of the activities and its effects on collaboration*”;
- **Cooperation** as “*the way the users interact in the shared space with the shared artefacts synchronous and asynchronously.*”

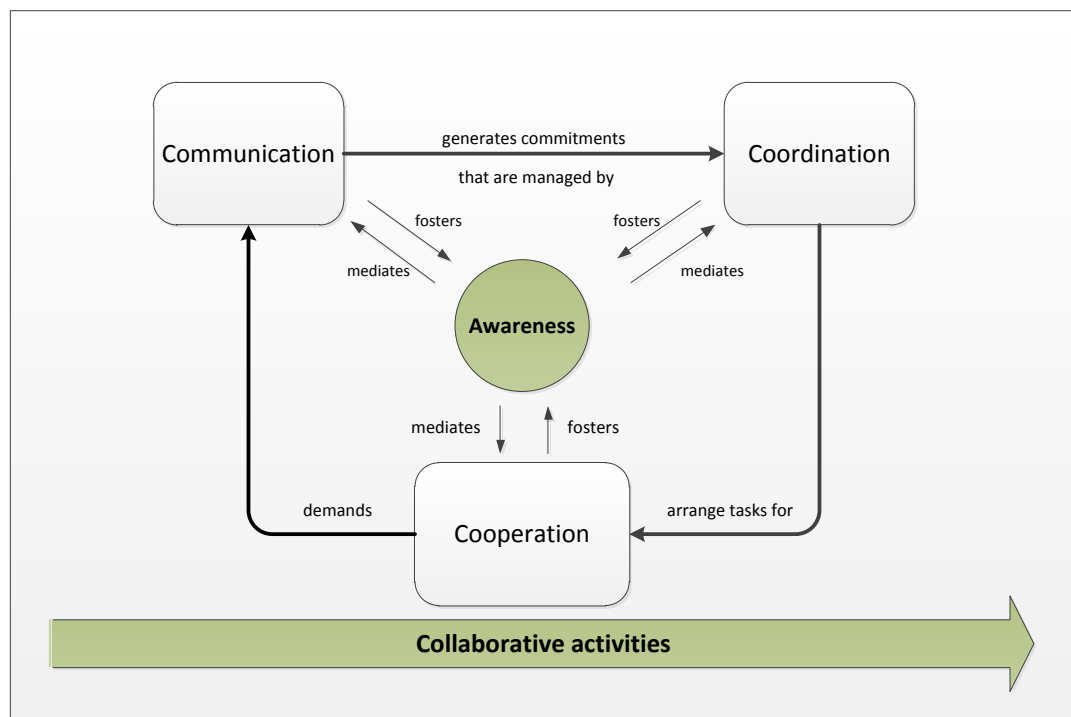


Figure 1: 3C collaboration model (adapted from Ellis et al. 1991; Fulks et al. 2008)

We will examine each of the three processes (communication, coordination and cooperation in turn and how each contributes to fostering awareness.

Communication has been identified as a fundamental aspect of Malone & Crowston’s (1994) interdisciplinary coordination theory where communication can be viewed as “*a way of managing producer/consumer relationships for information*”. Communication can be differentiated as formal and informal (Herbsleb & Mockus

2003; Kraut & Streeter 1995). Where informal communication is viewed as face-to-face discussions, telephone conversations with an emphasis on *ad hoc* communication and formal communication would be specifications, status review meetings, formal meetings and documentation. Pikkarainen et al., (2008) reported in collocated settings agile practices can improve both informal and formal communication. Sarker and Sarker (2009) suggest that “*communicative relationship agility*” in distributed environments is essential is where distributed teams feel a continuous awareness (and visibility) of distributed team members.

Kraut & Streeter (1995) advocate in software development effective coordination requires that different people working on a project agreeing on a common definition of their goals as well as sharing information and coordinate their activities. Cao & Ramesh (2007) recommend that agile environments call for coordination mechanisms that are different from the traditional development settings. Espinosa et al., (2007) suggests that effective coordination is enabled through shared knowledge of the task (task awareness) and shared knowledge of the teams. Team members of globally distributed teams who communicate via email have considerable difficulty in accessing, communicating and retaining information about remote colleagues and local contexts. This can lead to conflict and blame regarding project performance. Yap (2005) reported that frequent use of remote pairing sessions to create a “group mind” helped to increase cooperation, collaboration on ideas and removal of misunderstandings.

In collocated agile environments, there is an emphasis on constant cooperation between team members for the project tasks to progress (Beck & Andres, 2004; Ramesh et al., 2006). To alleviate the problems associated with the cooperation process between agile distributed teams, a number of researchers have proposed multiple communication modes for enhancing communication channels (Layman et al., 2006; Sutherland, 2007; Abbattista et al., 2008; Prikladnicki et al., 2012).

2.3 Related studies

There have been limited studies of assessing awareness within existing practices of distributed development environments. Herbsleb (2005) argued that to recognise the problems faced by developers in a distributed software development we need to further understand “*behaviour of software engineers, development team and*

organisations". Gutwin et al., (2004) studied how distributed open-source developers maintained group awareness and reported the main mechanisms to gain awareness were simple text communication channels. Damian et al., (2007) examined how the lack of awareness related to work items can cause communication breakdowns within global software teams. Recently, Smite et al., (2012) applied a modified version of the Gutwin & Greenberg's (2002) workspace awareness framework; where elements of Scrum methods were being applied in distributed environment. They reported best practices and tools proposed in related research are still not widely used in practice and there are multiple challenges in spreading awareness in cross-site settings.

Building on this line of research, as awareness is contingent to the processes of communication, coordination and cooperation this paper proposes using the 3C Collaboration model as an evaluative mechanism in an existing practice of distributed agile development environment. To seek a further understanding of how awareness is raised and maintained and comprehend what on-going challenges the distributed teams face.

3.0 Research Design

The research presented in this paper is based on a case study since the phenomenon examined occurs in a natural setting. The focus is on contemporary events and the study was designed to investigate the processes by which the distributed agile team members established awareness, how they maintained it and what on-going challenges they faced to sustain it. An interpretative approach considered appropriate where the focus is on contemporary events and where it is necessary to consider the context of the study. Such an approach is suitable where there is a complex relationship between people and processes and where researchers are interested in a 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. The unit of analysis in this case study are the individual team members (individual actors) and also the distributed teams (group of actors) which interact, communicate, coordinate and cooperate with each other to raise awareness within the teams and team members.

3.1 Contextual Setting

The case-study is based on an international bank and financial services company 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. 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 (BA), Technical Lead (Tech Leads), six senior software developers and two (Quality Assurance) 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 was responsible for developing a backend central depository system called the Operational Data Cache (ODC) system. The core business objective of ODC system is to ensure that all the divisions with the bank are using a single mandated “*golden copy*” of all the transactions from the front office systems. Within the bank the team is known as the ODC team - one team in two locations separated by time and space. Therefore the ODC team does not have upfront users or customers but a set of data feeds from various divisions which the ODC system would validate and ensure that everyone using the same copy of the data and safeguarding consistency of the data.

3.2 ODC Team Background

From the inception of the ODC project, the team adopted the tenets of XP with TDD (Test Driven Development) providing 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 to varying degrees. Some of team members in London had a similar IT background having previously worked as consultants 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; they were therefore mentored and exposed to XP practices by the London team.

The division of work is based on having software developers at both locations working together on 2 week iterations as one distributed team. Where the team’s

workflow is managed by an issue tracking system called JIRA (Altassian – see Figure 2). It is a web-based tool which allows remote team members to review the backlog and update tasks wherever they are. The JIRA system holds all the user stories providing an understanding of requirements to be completed and status of the outstanding user stories for the team at both locations (London and India). The research began approximately two and half years after its inception.

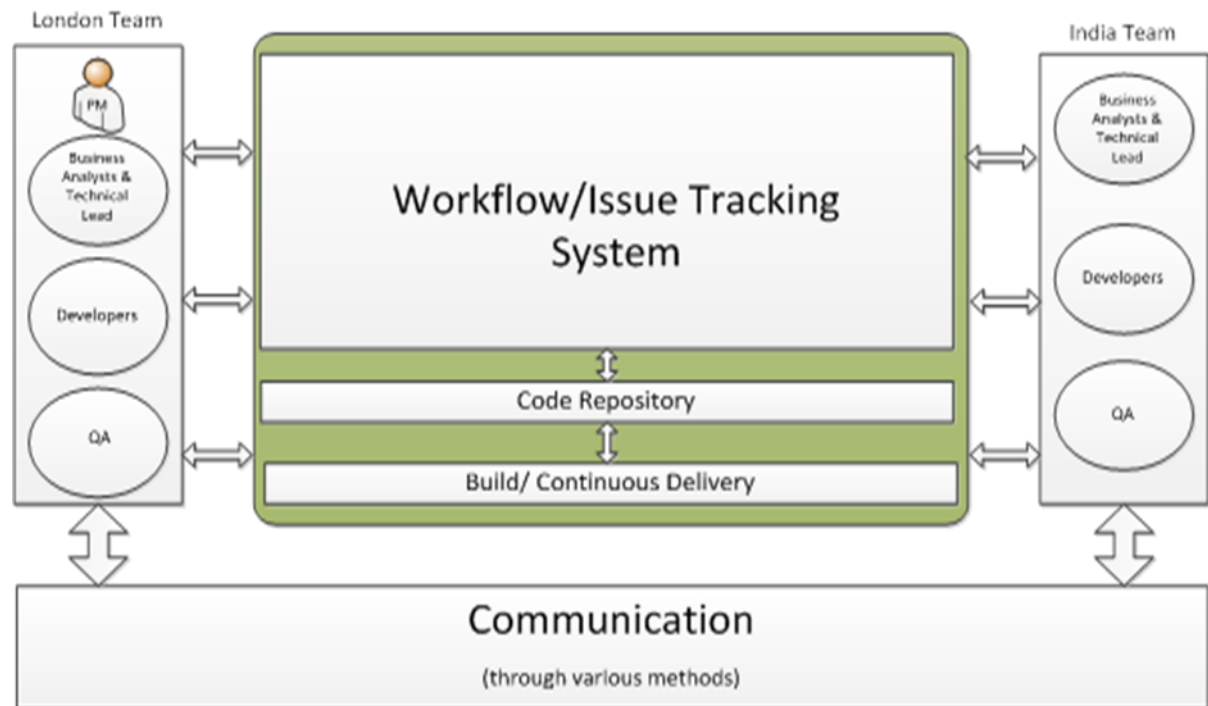


Figure 2: Workflow/Issue Tracking system

3.3 Data collection

Data collection involved sixteen semi-structured interviews each of which lasted, on average, one hour. Eight face-to-face interviews with individuals from the London team and eight video-conference interviews with the India team counterparts were completed. 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:

1. How awareness is raised and maintained within the two teams based in London and India in terms of communication, coordination and cooperation in both locations and how these actions enabled collaborative activities to take place.

2. The challenges related to developing and maintaining awareness faced at both locations.

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.

3.4 Data Analysis

Data analysis techniques involved transcribing all the interview findings in full and then uploading the textual data into qualitative data analysis software QSR-NVIVO software. The data from each interview was read and re-read and analysed through the main themes from the literature sources using a process called thematic coding (Fereday, 2006). Each data unit was a sentence or a short paragraph depending on the participant's conveyed meaning; this information was coded into themes (high-level codes) and sub-themes (lower order codes). The process gave an overall sense of how the dimensions of communication, coordination and cooperation overlap to raise awareness and increase collaborative activities within a distributed agile environment.

4.0 Case study findings

All the evidence from the case study is presented using the aspects of awareness as discussed in the 3C Collaboration model namely, communication, coordination, cooperation as headings. Although the findings are presented theoretically in separate categories, it is acknowledged that there are overlaps and the conceptual categories are inter-related.

4.1 Communication

To build and maintain awareness within agile teams, both formal and informal communication is a key activity. In collocated agile teams, this happens frequently face-to-face as per the agile principles (P4 and P6) in an open space environment (Fowler & Highsmith 2001; Beck & Andres 2004). In a distributed agile environment, the temporal distance can be the cause of number of problems (Carmel & Agarwal 2001; Ramesh et al., 2006). Formal communication could be thought of as: various meetings – daily stand-ups, review and planning meetings. Informal communication would be spontaneous conversations team members would have in order to clarify project tasks. The daily stand-up is an important practice as it gives an opportunity to

identify any problems and issues that have be resolved to ensure progress and is a useful way of exchanging information within a team. Within the ODC team, each location has its own separate daily stand-up meetings due to the time differences, but they also have a joint conference call meeting at 8:00am, where senior members of team from London and India attend.

“Communication is very awkward...with the time zones....Indian employers are leaving the office 11:30 UK time....Having been over there it, it’s infuriating, if you want to talk to someone in London about something and you know full well they’re in bed, because of time difference. So it’s maximizing those hours in the overlap....You can do a certain amount with email but it is not as good...when we are doing our analysis, we tend to write out user stories...we have these sessions, once a week we sit down and it’s a phone call meeting...people from London and from India, ...go through a set of stories....We always get development and QA involvement in there as well. Unfortunately, that part tends to be represented by people in London, but we do it on the phone, so that encourages people from India as well... we [inform] people what’s coming... down for development in the next iteration.”

“In the iteration itself, when we following a very good form of agile discipline, we have had sessions where developers have come together and had meetings and...had a good chat about...the stories, thoughts about development approach etc. Unfortunately that disintegrated into [India-London] split meetings”

From the quote it can be seen that communication is difficult due to the different time zones and therefore they try to make most use of the overlap hours to establish awareness within distributed team of the project tasks.

Additionally the team have introduced another process of formal communication where the developer has to demonstrate that the user story requirements are met; this was carried out by conference call meetings which raised awareness between all the relevant people within the team and the user before being passed on to the QA team for testing:

“...before we allow it to go into test, the developer must demonstrate what they have built and show it is working and there is nothing obvious that is broken. And those again are phone call meetings, with shared screens. So...the developer might be India based, the QA might be India based, but the BA is over here, and also there are certain people who depending on what part of the system it is...who are technical lead, so we’d make sure they’re involved and sometimes the project stream managers are also interested in seeing that as well.”

In collocated settings, the focus on the informal communication maintains the awareness of tasks in terms of changes that may have taken place. The ODC team use Instant Messaging (IM) chat channels, emails, and telephone to communicate on a daily basis. The ODC team have different channels such as the Development Channel and a Test Channel are used for informal conversations to maintain awareness of what is happening as illustrated in this quote:

“...of what is happening and who is working on which part of the code or if something goes down then we have continuous integration process which builds the entire code every time someone checking so if something goes wrong that or with testing environments – then we tend communicate over the common chat. So that chat happens on a daily basis we will see about 30 to 40 entries every day. We sometimes do a group video conference call – I don’t remember the last time we did – we haven’t done that recently. I believe we should do it more often.”

From the quotes it can be seen how ODC project team employ various formal and informal communication processes to maintain awareness and try achieve the agile principles (P4 & P6) as much as possible. However, due to temporal distance and practices have been adapted and tailored to suit the distributed settings. The informal communication is happening quite frequently to maintain task awareness, but the developers put forward need for having a visual presence.

4.2 Coordination

In collocated agile environments user stories are project artefacts which represent requirements for the development team and aid communication and coordination. They represent “units of functionality” and are displayed in an area of physical space called the “informative workspace” or the “story wall” and they are organised in terms of status: open, in-progress, completed (Beck & Andres 2004; Sharp & Robinson, 2008). Cockburn (2007) refers to this workspace as an ‘information radiator’ which has the characteristics of reporting information changes over time; it is visible to the software development team and shows the work-breakdown for each increment. Team meetings usually take place near the story wall to encourage face-to-face communication and coordination, reporting on any problems within the context of the project. In ODC team they use the JIRA system as a workflow system giving team members a real-time visibility of documents the progress of current user stories. It creates a shared platform for both locations and a “*virtual wall*” which can help in the coordination and generation of appropriate workflows. It also allows business analysts and project managers to re-prioritise tasks and offers management reporting tools:

“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 in JIRA]”.

The JIRA system also allows team members to keep track of changes and modifications made to particular functionality this ensures and awareness is maintained.

“I think if there’s a bit functionality that I’m interested in, and the analysis is being done in India for instance of the features you can use with JIRA is that you can watch it and you will get an email if it changes and I can look where we write stories. I can get a reminder, dip into it and see what they have changed, just to keep me up to speed with that area of functionality.”

In agile methods, maintaining team and task awareness is crucial as the iterations are short (usually two weeks) and having a platform to represent the current user stories is vital as per agile principle P5. From the above, it can be seen that the ODC team have successfully adapted the “story wall” into a “virtual wall” where user stories are organised in terms of status and can be seen by all team members (Beck & Andres, 2004; Sharp & Robinson, 2008; Berczuk, 2007).

4.3 Cooperation

In collocated XP environments, the practice of pair-programming allows developers to brainstorm, clarify ideas with other team members and keep track on the progress of the task. It also reduces the overall socio-cultural distance between team members and reinforces the sense of team cooperation and the one team mind-set (Beck & Andres; 2004). When the ODC team was set up there was a definite intention of one team based in two locations which would be underpinned by the communication and coordination processes described above. However, developing cooperation between the two locations was found to be one of the most challenging for the ODC team. As the Indian offshore team was less familiar with Agile XP approaches and the London team mentored and supported them to gain a better shared understanding of the project by using abridged pair-programming sessions as a vehicle to develop cooperation, enhance programming skills and 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 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:

“... 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.”

Although this type of pair-programming activity where the developers were working on the same user stories was supposed to support the Indian developers and generate a sense of cooperation between team members, it initially created considerable misunderstandings which caused frustration and friction between developers:

“...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.”

It was only after the exchange visits took place that that teams felt more attached and cohesive bonds were formed through spending time at the other location:

“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.”

This highlights that the process of developing cooperation between the team members has only improved gradually overtime after the exchange visits took place and knowledge sharing has taken place though the local-global interfaces (Abbott et al., 2012; Sahay et al., 2003).

Another activity to encourage one team mind-set and cooperation was the ethos of not allocating user stories to either locations, but for the developers to pick the highest priority user story from JIRA which they felt competent to complete from previous experience.

“When a developer has finished on project they go on to pick another story, and they go onto pick whatever is the highest priority, and work on that basis. The development of the user story - you don’t know where it’s going to happen, someone could pick something up in India, someone could pick something up in London, and it’s not allocated across the locations.”

From the above examples we can see that fostering cooperation between the two locations has been arduous task and it has not been an easy endeavour to achieve the agile principle (P5). The exchange visits and staff rotation removed some of the barriers, as after the visits, team members felt they were not talking to strangers but their team counterparts in another location. This subsequently increased cooperation and cohesiveness within the team and raised the shared team awareness.

4.4 Awareness

From the above it can be seen that the ODC team engage in a number of different processes and practices to foster and maintain awareness within the distributed team.

Additionally, we can see that some of the agile practices have been modified and adapted for the distributed setting for the project tasks to progress. However, it can also be recognised that having a number of different tools and technologies to support awareness was not enough for cooperation to take place between offshore and onshore team members. Cooperation between the team members evolved once trust was established from the exchange visits and staff rotation process. Some team members felt that awareness could be further improved by increasing the visual presence of the offshore team, use web cameras and use of video walls. To visualise workflows, they suggested the use of virtual Kanban boards or virtual whiteboards. These suggestions emphasise the importance of raising awareness through “visual cues” which are a natural part of collocated XP teams (Sharp & Robinson, 2008; Fussell et al., 2000). Table 2 shows the three dimensions of the 3C Collaboration model: communication (formal & informal), coordination and cooperation against the recommended agile practices relating to agile principles P4, P5 and P6. This is compared against the ODC team adapted practices. From this table we can easily identify the indicators of potential loss of awareness by discerning where there are changing practices within the team. For example, within the ODC team where some practices started off jointly but are now held separately in the two locations such as the planning meetings, these could be brought to attention as indicators for the potential of loss of awareness which could be explored further.

<i>3C- Collaboration Dimension</i>	<i>Agile Practices related to Agile Principles (P4, P5, P6)</i>	<i>ODC team adapted practices</i>
<i>Formal communication</i>	Daily meetings Weekly planning and review meetings, current iteration meetings Developer demos user story to BA and user before passing it on to QA	<ul style="list-style-type: none"> ▪ Daily 8:00 am conference call between London & India (attended by PM,BAs, Tech. Leads and Senior Devs. & Senior QA) ▪ Location based as well ▪ Used to be carried out jointly, however now carried out separately at each location and then follow up conference calls ▪ Using screen-sharing software and conference calls
<i>Informal Communication</i>	Day to day developer conversations for clarifying tasks 5 min user story chat (to check if the story is up-to-date)	<ul style="list-style-type: none"> ▪ Using IM, email and phone, (abandoned use of video conference) ▪ Carried out separately at each location
<i>Coordination</i>	View user stories and the wall Changes made to user stories	<ul style="list-style-type: none"> ▪ Using Issue tracking system called JIRA ▪ Using JIRA

Cooperation	Progress made on user stories Planning two week iterations (adding new user stories) Using the same code repository	<ul style="list-style-type: none"> ▪ Using JIRA ▪ Using JIRA ▪ Integrated via JIRA
	Developing one team mind-set	<ul style="list-style-type: none"> ▪ Frequent visits and staff rotation happened at the inception of the project (recently stopped due organisational budgets). ▪ User stories are not allocated to particular location or particular developers. The ethos of ODC team is for the developer to pick the user story with the highest priority which he feels can complete.
	Pair programming (PP)	<ul style="list-style-type: none"> ▪ Carried out initially across locations, to support India team members but this caused frustration due to language barriers and screen-sharing software did not allow for concurrent editing. ▪ Now PP is carried out within each location depending on the user story.
	QA-Dev. Team relationship	<ul style="list-style-type: none"> ▪ Majority of the QA team is based in India. Indian developers and testers work more closely and have greater cooperation. The London developers feel that sometimes misunderstandings occur as they are not in the same location as the majority QA team.

Table 2: 3C-Collaboration dimension against ODC team adapted practice

5.0 Discussion

In this section we return to our research questions (1) “*How do distributed cross-cultural agile teams raise and maintain increased awareness for collaborative activities to take place across the boundaries?*” and (2) “*What are the on-going challenges that distributed agile teams face to sustain awareness?*” and discuss how the case study gives us a better understanding of these issues. We utilised the 3C Collaborative model as an evaluative mechanism and investigated its dimensions within the case study. The findings reveal important insights into how the ODC team works in practice and employs a number of strategies which aid to raising and maintaining awareness.

5.1 Raising awareness through communication

The concept of awareness is contingent on communication, coordination and cooperation and in this case study the process of communication can be seen as an essential prerequisite for the coordination and cooperation processes to take place. If the formal or informal communications are hampered or restricted in anyway it can

cause potential impediments in raising awareness (Espinosa et al., 2007; Holmstrom et al., 2006; Sarker & Sahay 2004). One of the strategies employed by the ODC team to reduce the potential loss of awareness is to make effective use of the overlap hours, whereby formal and informal synchronous communications take place to ensure that awareness is continuously being maintained. Yet, there are times when communication breakdowns do occur and awareness deteriorates especially where developers have to handle a multiplicity cultural and technological barriers, as well as coping with differences of knowledge and skills of their team counterparts at the offshore location (Damian et al., 2007). This was illustrated in our case when cross-border pair-programming activities took place. This highlights that awareness support tools are clearly not enough in supporting the joint complex activities and where socio-cultural distance plays a greater emphasises on raising awareness within the team.

5.2 Raising awareness through coordination

From a coordination perspective, it is evident that it requires far greater effort to raise and maintain awareness in globally distributed environments rather than collocated environments (Lee et al., 2006; Cataldo et al., 2007, Damian et al., 2007). From our case-study, the ODC team deploy various set of technologies, tools (such as JIRA, screen sharing software, IM, email and conference calls) for awareness to be raised and maintained and to keep the progress of project workflow. Some coordination tools work very well such JIRA and are suited for such an environment; however others as such as screen-sharing software can sometimes cause developer frustrations. Unfortunately, at present, there is not one single independent awareness support tool which encapsulates all the requirements needed for agile distributed development teams; organisations therefore have to prioritise their collaboration needs and their tools to support them (Lanubile et al., 2010; Prikladnicki et al., 2012).

5.3 Raising awareness through cooperation

Within the ODC team fostering awareness through the process of cooperation was most challenging, as it required the team members to develop a rapport over the telephone and work together on one piece of functionality or user story across locations and this initially caused a several misunderstandings and tensions. This type of distributed pair-programming has been reported as “*hindering cooperation and the*

one team goal” (Yap, 2004). Kircher et al. (2001) additionally reported that even with the aid of video-conferencing and other synchronous tools this type of remote pairing could not completely substitute the “*physical closeness*” as effectively as in a truly agile environment. Additionally, this type of distributed pair-programming practice also contrasts with other distributed agile development studies where the onshore team allocates the user stories to be completed by the offshore team (Abbott et al. 2013). Hence, it was only after the exchange visits and staff rotation that the ODC team gradually felt like “one team” and the cooperation with their counterparts improved. As these visits raised team awareness of the offshore team, allowed the cultural barriers to be subdued and allowed a “negotiated” culture to develop for over time (Krishna et al. 2004, Kotlarsky & Oshri, 2005). On reflection the team members appreciate that the joint process of remote pair-programming and the visits has augmented the domain knowledge and programming skills of the India team. But now developers prefer to carry out these activities within each location as raising awareness for closely knit activities without the close proximity was is arduous task.

5.4 On-going challenges to sustain awareness

In response to our second research question, the case study illustrates sustaining awareness across the locations is a continuous process especially as timeframes for iterations to deliver software are short (usually two weeks) and time-zone differences can delay in solving the problems quickly. Therefore the ODC team have tried to construct a truly agile environment as far as possible, but with an appropriation of the agile principles (P4, P5 and P6) such that it is an “*agile way of working*” with a number of modified practices to suit the context of a distributed environment. As in this setting, continually raising and sustaining awareness is much more difficult to achieve compared to a collocated environment. Additionally, some of the team members wished for an increase of awareness by “*visual cues*” so that there was some presence of their counterparts by use tools such as video walls and webcams (Sarker & Sarker 2009).

This also carries implications for senior management of the ODC team as the needs of globally distributed agile teams are far greater than traditional GSD teams. As traditional GSD teams usually plan-based methods are applied with a sequential approach where the project requirements would be passed on to the offshore team and the software would be released towards the end of project lifecycle. Whereas in

distributed agile environments there is a constant interplay between the offshore and onshore teams to achieve the project goals and this causes an essential prerequisite for raising and maintaining awareness. When practices are hindered by lack of awareness it can cause impairments to project goals.

6.0 Conclusions & Further work

Distributed agile development is receiving extensive attention among the GSD domain and researchers. The key reason being that in collocated settings agile methods have proved to deliver working software in an incremental manner compared to traditional plan-driven methods and now these methods are being espoused and adapted in globally distributed settings.

This paper offers a two-fold contribution, towards the literature in the area of distributed agile practices. Firstly, by adopting the 3C Collaboration model, as an evaluative mechanism to assess the processes separately, we can verify they all have to be in synergy for awareness to be continually fostered in distributed agile settings. Additionally, the model can identify where there is potential loss of awareness which can be subsequently investigated further in order to see if any of distributed team activities are suboptimal. The findings have allowed us to see how constant awareness is an indispensable requirement for distributed agile teams and consequently how it supports them in their collaborative practices. Our analysis helps us better understand how agile practices are moderated and managed to form an “*agile way of working*” within a globally distributed setting. Secondly, the implications for senior management planning to use the distributed agile mode in a global setting should be attentive to be more responsive in developing numerous strategies to raising and maintaining awareness, as it can “*make or break*” the success of a project within distributed cross-cultural agile teams.

The insights discussed here are based on one case study based within a financial context, which is a limitation of this paper. Further research is required among other distributed agile teams, to see if there are any similarities and comparability within cases which can lead us to a deeper understanding of how raising awareness to enable successful collaborative practices. To strive for additional evidence we seek to identify and establish awareness patterns between different kinds of distributed agile environments. Additionally, examine the connections between awareness and how a shared understanding is developed between the onshore and offshore team.

Acknowledgements: The authors would like to express their gratitude to the industrial collaborator for their support and the participants for their time.

References

- Abbattista, F., Calefato, F., Gendarmi, D., Lanubile, F. (2008), *Incorporating social software into distributed agile development environments*, ASE Workshops 2008, 46–51.
- Abbott, P., Zheng, Y. and Du, R., Innovation through collaborative partnerships: creating the MSN News for iPad app at VanceInfo Technologies, *Journal of Information Technology Teaching Cases* (advance online publication)
- Abbott, P.; Zheng, Y.; Modi, S.; Du, R. (2012), *Moving up the Value Chain : Exploring the Issues for Chinese Software & Services Outsourcers*. PACIS 2012 Proceedings.
- Agerfalk, P.J., Fitzgerald, B. (2006) *Flexible and Distributed Software Processes : Old Petunias in New Bowls ?* *Commun. ACM* 49,26–34.
- Ambler, S. (2002). *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*, John Wiley & Sons.
- Batra, D. (2009) *Modified agile practices for outsourced software projects*. *Commun. ACM* 52, 143–148.
- Beck, K. and Andres, C. (2004) *Extreme Programming Explained*, 2nd ed, Addison-Wesley
- Benbasat, I; Goldstein, D. K.; Mead, M. (1987), *The Case Research Strategy in Studies of Information Systems*, *MIS Quarterly*, 11(3), 369–386.
- Biehl, J.T., Czerwinski, M., Smith, G., Robertson, G.G. (2007) *FASTDash: a visual dashboard for fostering awareness in software teams*, in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. pp. 1313–1322.
- Berczuk, S. (2007), *Back to Basics: The Role of Agile Principles in Success with a Distributed Scrum Team*, in: *Agile Conference (AGILE)*, 2007, 382–388.
- Carmel, E., Agarwal, R. (2001) *Tactical approaches for alleviating distance in global software development*. *IEEE Software* 18, 22–29.
- Cao, L., Mohan, K., Xu, P., Ramesh, B. (2009) *A framework for adapting agile development methodologies*. *European Journal of Information Systems* 18, 332–343.
- Cao, L. Ramesh B., "Agile Software Development: Ad-hoc practices or sound principles?" *IEEE IT Professional*, Vol. 9, (2), 2007 pp. 41–47.
- Carmel, E., Tjia, P. (2005) *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press.
- Cataldo, M., Bass, M., Herbsleb, D., J. and Bass, L. (2007) *On Coordination Mechanisms in Global Software Development*. *International Conference on Global Software Engineering*. 71–80.
- Cockburn, A. (2002), *Agile Software Development*, Addison-Wesley.
- Cramton, C.D. (2001) *The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration*, *Organization Science* 12, 346–371.
- Damian, D., Izquierdo, L., Singer, J., Kwan, I. (2007) *Awareness in the wild: Why communication breakdowns occur* in: *Global Software Engineering, 2007. ICGSE 2007*. pp. 81–90.
- Dourish, P., Bellotti, V. (1992) *Awareness and coordination in shared workspaces*, in: *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work, CSCW '92*. ACM, New York, NY, USA, pp. 107–114.
- Dullemond, K., Van Gameren, B., Van Solingen, R. (2009) *How technological support can enable advantages of agile software development in a GSE setting*, in: *Global Software Engineering, 2009. ICGSE 2009. IEEE International Conference* pp. 143–152.

- Ellis, C.A., Gibbs, S.J., Rein, G. (1991) *Groupware: some issues and experiences*. Commun. ACM 34, 39–58.
- Espinosa, J.A., Slaughter, S.A., Kraut, R.E., Herbsleb, J.D.(2007) *Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development*. Organization Science 18, 613–630.
- Fereday, J. (2006). *Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development* | |. International Journal of Qualitative Methods, 5(1), 80–92.
- Fitzgerald, B., Hartnett, G., Conboy, K.(2006) *Customising agile methods to software practices at Intel Shannon*. European Journal of Information Systems 15, 200–213.
- Fowler, M., & Highsmith, J. (2001), The agile manifesto. <http://www.agilemanifesto.org> accessed 20.07.2012
- Froehlich, J., Dourish, P. (2004) *Unifying artifacts and activities in a visual tool for distributed software development teams* in: Proceedings of the 26th International Conference on Software Engineering. pp. 387–396.
- Fuks, H., Raposo, A., Gerosa, M.A., Pimentel, M., Filippo, D., Lucena, C.(2008) *Inter- and intra-relationships between communication coordination and cooperation in the scope of the 3C Collaboration Model*, in: 12th International Conference on Computer Supported Cooperative Work in Design, 2008, (CSCWD) pp.148 –153.
- Fussell, S.R., Kraut, R.E., Siegel, J., 2000. Coordination of communication: Effects of shared visual context on collaborative work, in: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work. pp. 21–30.
- Gutwin, C., Greenberg, S. (2002) *A descriptive framework of workspace awareness for real-time groupware*. Computer Supported Cooperative Work (CSCW) 11, 411–446.
- Gutwin, C., Penner, R., Schneider, K. (2004) *Group awareness in distributed software development*, in: Proceedings of the Conference on Computer Supported Cooperative Work, (CSCW) ACM, New York, NY, USA, pp. 72–81.
- Herbsleb, J.D.& Mockus, A.(2003) *An empirical study of speed and communication in globally distributed software development*. IEEE Transactions on Software Engineering 29, 481 – 494.
- Herbsleb, J.D. (2005) *Beyond computer science*, in: 27th International Conference on Software Engineering, 2005. (ICSE) Proceedings, pp. 23 – 27.
- Herbsleb, J.D. (2007) *Global Software Engineering: The Future of Socio-technical Coordination*, in: *Future of Software Engineering* Presented at the Future of Software Engineering, 2007. FOSE '07, pp. 188 –198.
- Holmström, H., Fitzgerald, B., Ågerfalk, P.J., Conchúir, E.Ó., (2006). Agile Practices Reduce Distance in Global Software Development. Information Systems Management 23, 7–18.
- Kircher, M., Prashant, J., Corsaro, A., & Levine, D. (2001), Distributed eXtreme programming. in Proceedings of the Conference on XP2001 eXtreme Programming and Flexible Processes in Software Engineering.
- Ko, A.J., DeLine, R., Venolia, G.(2007) *Information Needs in Collocated Software Development Teams* in: Proceedings of International Conference on Software Engineering, ICSE '07. IEEE Computer Society, Washington, DC, USA, pp. 344-353.
- Kotlarsky, J., & Oshri, I. (2005), Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. European Journal of Information Systems, 14(1), 37-48.
- Kraut, R. E., & Streeter, L. A. “*Coordination in Software Development*,” Communications of the ACM (38:3), 1995, pp. 69-81.

- Krishna, S., Sahay, S., Walsham, G. (2004) *Managing cross-cultural issues in global software outsourcing* Commun. ACM 47, 62–66.
- Lanubile, F., Ebert, C., Prikladnicki, R., Vizcaino, A. (2010) *Collaboration Tools for Global Software Engineering* IEEE Software 27, 52 –55.
- Layman, L., Williams, L., Damian, D., Bures, H. (2006), *Essential communication practices for Extreme Programming in a global software development team* Information and Software Technology 48, 781–794.
- Lee, G., DeLone, W., Espinosa, J.A., 2006. *Ambidextrous coping strategies in globally distributed software development projects*. Commun. ACM 49, 35–40.
- Malone, T.W., Crowston, K.(1994) *The interdisciplinary study of coordination*. ACM Comput. Surv. 26, 87–119.
- Nerur, S., Mahapatra, R.K., Mangalaraj, G.(2005) *Challenges of migrating to agile methodologies*. Communications of the ACM 48, 72–78.
- Omoronyia, I., Ferguson, J., Roper, M., Wood, M. (2010) *A review of awareness in distributed collaborative software engineering*. Software: Practice and Experience 40, 1107–1133.
- Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., Still, J. (2008) *The impact of agile practices on communication in software development*. Empirical Software Engineering 13, 303–337.
- Prikladnicki, R., Marczak, S., Carmel, E., Ebert, C. (2012), *Technologies to Support Collaboration across Time Zones*. IEEE Software 29, 10 –13.
- Ramesh,B.,Cao,L., Mohan, K., Xu, P. (2006) *Can distributed software development be agile?* Commun. ACM 49,41–46.
- Sahay, S., Nicholson, B. and Krishna, S. (2003). *Global IT Outsourcing: Software Development Across Borders*. Cambridge University Press, Cambridge.
- Sarker, S., Sahay, S. (2004) *Implications of space and time for distributed work: an interpretive study of US-Norwegian systems development teams*. European Journal of Information Systems 13, 3–20.
- Sarker, S., Sarker, S. (2009) *Exploring Agility in Distributed Information Systems Development Teams: An Interpretive Study in an Offshoring Context*. Information Systems Research 20, 440–461.
- Sarma, A., Noroozi, Z., Van Der Hoek, A.(2003) *Palantír: raising awareness among configuration management workspaces*, in: Proceedings. International Conference on Software Engineering, Portland, Oregon, USA, 2003, pp. 444-454.
- Sharp, H., Robinson, H. (2008) *Collaboration and co-ordination in mature eXtreme programming teams* International Journal of Human-Computer Studies 66, 506–518.
- Smite, D., Dingsoyr, T. (2012) *Fostering Cross-site Coordination through Awareness: An Investigation of State-of-the-Practice through a Focus Group Study*, in: 2012 EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), pp. 337 –344.
- Steinmacher, I., Chaves, A., Gerosa, M.(2010)*Awareness support in global software development: a systematic review based on the 3C collaboration model*. Collaboration and Technology 185–201.
- Storey, M.A.D., Cubranić, D., German, D.M.(2005)*On the use of visualization to support awareness of human activities in software development: a survey and a framework*, in: Proceedings of the 2005 ACM Symposium on Software Visualization.pp.93–202.
- Sutherland,J.,Viktorov,A., Blount,J. and Puntikov,N.(2007)*Distributed scrum: Agile project management with outsourced development teams*. Proceedings of HICSS 2007, 274-274

- Walsham, G. (2006), *Doing interpretive research*. *European Journal of Information Systems* 15, 320–330.
- Yap, M., (2005) *Follow the sun: distributed extreme programming development*, in: *Agile Conference, 2005. Proceedings*, pp. 218 – 224.
- Yin, R. K. (2003), *Case study research, design and methods*, 3rd ed, Sage Publications.