Remote Working and Collaboration in Agile Teams

Completed Research Paper

Advait Deshpande
The Open University
Walton Hall, Milton Keynes, UK
advait.deshpande@open.ac.uk

Helen Sharp
The Open University
Walton Hall, Milton Keynes, UK
helen.sharp@open.ac.uk

Leonor Barroca
The Open University
Walton Hall, Milton Keynes, UK
leonor.barroca@open.ac.uk

Peggy Gregory
University of Central Lancashire
Preston, UK
ajgregory@uclan.ac.uk

Abstract

Agile software development relies heavily on tight and continuous collaboration, which becomes a challenge when team members work at a distance. Despite significant focus on distributed Agile working, remote working, when only one or two individuals are not co-located with the rest of the team, remains largely unexplored. We focus on one organisation with several such teams and investigate one in detail using distributed cognition – a theoretical framework for studying collaborative work. We present the results of a group retrospective, and a comparative analysis of collaboration in the team, taking the contrasting perspectives of the remote worker and his co-located teammates. The analysis shows substantial differences in three aspects: virtual artefacts; information flow; and the primacy of structure and facilities provided by collaborative platforms. Platforms that support meaningful collaboration and engagement for the remote worker, and create parity between all members of the team are crucial to integrating capability.

Keywords: Remote Working; Distributed Cognition; DiCoT; Agile

Introduction

The Agile Manifesto promotes the principle that face-to-face conversation is the most efficient and effective method of conveying information within a team (Beck et al, 2001). The communication and mutual awareness enabled by co-located teams provide the basis for essential features of the Agile approach such as collaboration, regular delivery of working software, and responding quickly to change. However, co-location is not always possible for various practical and business reasons, and this presents a particular challenge for Agile practitioners because of the heavy reliance placed by Agile on tight and continuous collaboration.

Research on non co-located teams has largely focused on 'distributed' teams, where sub-teams are sited in different locations, often in different countries. However, there are other team configurations. One is the 'dispersed' team (Sharp et al 2012), where team members work alone at different locations. Another is the 'hybrid' team (Staples and Webster 2008), which consists of some co-located members and some remote members who work alone. The use of remote workers is not unusual as businesses may opt to use home-based workers in order to keep expert staff or when recruitment is difficult. This paper focuses on hybrid teams where a minority of workers are remote and the rest of the team is co-located. It presents an investigation into the realities of remote working, specifically an in-depth study of one software company

with several hybrid teams each of which is largely co-located but with one full-time remote worker. The research study draws on a range of data collected from the company, including an in-depth distributed cognition analysis of one hybrid team in particular. The research questions for the research were:

RQ1: What kind of issues (if any) are faced by a hybrid Agile team in practice?
RQ2: How does a hybrid Agile team with a remote worker collaborate?
RQ3: How does the experience of remote working compare between a remote worker and his co-located teammates in a hybrid Agile team?

The next section summarises relevant literature, and the third introduces the case organisation. There were two elements to the study presented here: a group discussion retrospective and an in-depth distributed cognition analysis of one team. The fourth presents data from the group discussion, which addresses RQ1 and sets the scene for the main empirical study: an in-depth distributed cognition analysis of one team, which addresses RQs 2 and 3. The fifth section presents three aspects of the analysis of the remote worker and the office team based on the distributed cognition analysis. The sixth section discusses the findings, and limitations and the final section concludes the paper.

Background

There is a significant body of literature that focuses on global software engineering (GSE) for distributed Agile teams. Both advantages and disadvantages of this approach have been identified. A systematic review of Scrum practices in GSE (Hossain et al 2009) identified that the main difficulties were communication, collaboration, tool support and large teams. Many communication difficulties are linked to cultural and linguistic differences between teams. These are however compounded by other difficulties. Slow or unreliable transmission quality over communications networks can cause problems. Collaboration can be hampered by a lack of effective tools for managing activities such as task boards, bug trackers, and backlog tools. Teams distributed over more than two locations in different time zones experience particular problems. Jalili and Wohlin (2011) conducted a systematic review of literature on Agile GSE in order to identify which Agile practices were reported to have been successfully used in distributed Agile teams. The most commonly reported successes were stand-up meetings, sprints/iterations, continuous integration, sprint planning, retrospectives, pair programing, sprint review, Test Driven Development, and Scrum of Scrums. They also found that the majority of Agile GSE teams modified practices because of situational requirements. More recently, Rizvi et al. (2015) published another systematic literature review of distributed Agile software engineering where they identify the primary challenges that need addressing in four areas: communication, coordination, collaboration and culture. They also focus on the reasons and conditions that led to the adoption of distribution and highlight which Agile methods have been successfully adopted. Some of these observations are relevant for hybrid teams, but GSE has limited relevance for the discussion in this paper since the participants consisted of a co-located team and remote workers who all worked within a single location and time zone (United Kingdom and GMT/BST respectively), and used only one language (English) for all communication.

Hummel et al (2013) summarise published findings on several aspects of Agile communication including the effects of team distribution. Although communication is the key issue for distributed teams, the need for a well-defined customer is also influential for success. Several studies claim that the enforced communication practices of Agile development help to overcome some challenges for distributed teams and hence can reduce temporal, geographical, and socio-cultural distances (Holmström et al 2006; Paasivaara et al 2008). They also mention the issue of trusted relationships and a shared understanding as being important indicators. Recommendations for practice include the use of 'seeding visits' at the beginning and 'maintaining visits' during projects in order to build trust (Paasivaara et al 2008). Other suggestions include longer stays where a member of a team is physically located with a remote team. Any team that is not entirely co-located depends on electronic communication, and the use of tools is inevitable. Findings suggest that a diversity of communication modes (asynchronous and synchronous) is needed.

Dispersed teams are less well understood. Sharp et al. (2012) studied a partially dispersed team using distributed cognition analysis (Hutchins 1995) and compared this analysis with one conducted for co-located settings (Sharp et al 2006). In the co-located setting they found that the team relied on discussion

and did not use many mediating artefacts; that information flows were simple and open; and that the team worked in an environment in which information was plentiful and easy to access. Within the co-located setting information sharing was more implicit and the social structures were highly sophisticated. In contrast they found that the dispersed team almost exclusively used the virtual rather than the physical space; and hence relied on complex digital artefacts with sophisticated structures. When compared to the co-located team, the social structures were less sophisticated and simpler in the dispersed team. Team members therefore needed to be familiar with the tools being used; information sharing needed to be explicit; information was transformed more often than in a co-located setting; and individuals had to take responsibility for deciding what information to share, when and through which medium.

Virtual teams are often created specifically to bring together diverse expertise and knowledge, and hence team members need to rely on each other and share knowledge. Staples and Webster (2008), researching outside the software engineering domain, indicate that knowledge sharing in virtual teams may be substantially different than in co-located teams. Electronic communication is effective for sharing explicit knowledge but not for tacit exchanges which typically require close personal contacts. In virtual teams more of the knowledge being shared is of lower quality and less sensitive because it can be more difficult to share emotions, experiences and insights. The reduction in the quality of knowledge being shared can lower team performance and reduce members' intentions to remain on the team. Their findings indicate that in virtual teams increased knowledge sharing is associated with increased team effectiveness; but if the teams are unbalanced or hybrid there is reduced knowledge sharing, the trust within the team is weaker, and as an outcome the team effectiveness is lowered.

In a comprehensive discussion of distance work in general Olson & Olson (2014) identify four stubborn problems of distance: out of sight out of mind, trust, culture and time zones. They give recommendations to: members of, managers of, and organisations supporting distributed teams; they also categorise tool support into four types: communication, coordination, information repositories and computational infrastructure. It is striking, in this context, that they do not include a category for collaboration.

The Case Organisation

The case organisation is Workplace Systems Ltd., a Milton Keynes-based company that specialises in workforce management software and employs primarily Scrum-based Agile practices. Workplace Systems Ltd. is keen to utilise the capability of remote workers i.e. team members who live beyond a reasonable commuting distance from their main office. Full-time remote workers are present in two of their teams and one developer works remotely for part of the week and comes into the office for the remainder of the week. All teams follow a release cycle of 8 weeks in which 3 sprints of 2 weeks each are focused on the product backlog, and the final sprint (i.e. the last 2 weeks) is mostly devoted to making fixes and regression testing. The key Agile practices used are: backlog grooming and planning poker, daily standups, sprint planning, show & tells, and retrospectives.

Our data gathering involved around 20 members of the software development staff, including an in-depth study of one team comprised of one Scrum master, two developers (one of whom was the remote worker), two part-time testers, and a product owner shared with other teams. The product owner was not available as a participant in the study. Table 1 presents a list of the key software tools used by the team members (co-located and remote) along with the tools' purpose and a brief description of how the team used each tool. These tools are commonly used by software development teams to support communication, collaboration and co-ordination as well as the software development task itself.

The product backlog was agreed by all members of the team in the backlog grooming meeting, based on the user stories and the number of story points agreed to be delivered in a release. The sizing of user stories was agreed in a prior, separate meeting via planning poker using the Hatjitsu online system (see Table 1). Both meetings were usually chaired by the Scrum master.

The daily standup took place at 9:15 in the morning and focused on team activities completed on the previous day, the work to be done today, and any blockers. The team used a physical Kanban-style board for keeping track of work during standups, and an online system, Jira, to maintain all data. The physical board mirrored the data on Jira. Jira was the main source for the most up-to-date information.

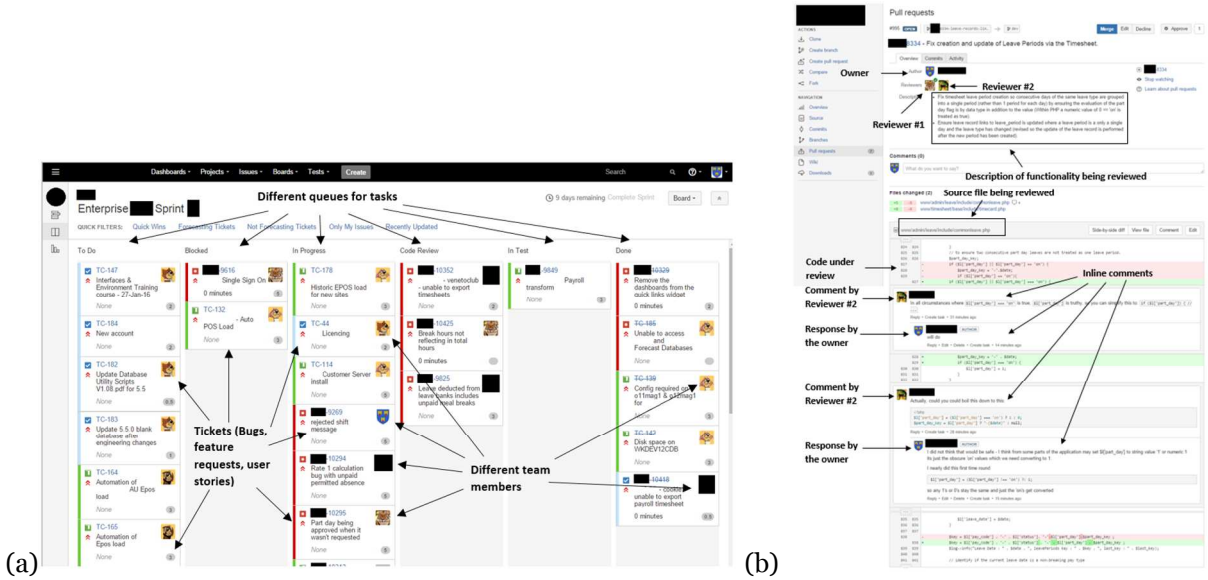
The retrospectives similarly followed a combination of online and offline practices. The co-located office team worked with post-it notes in a meeting room and discussed the outcomes of a release while the remote worker participated via a video or audio call, or through an online support tool such as Appear.in. All project teams in the company held a combined "show & tell" with relevant stakeholders once for each release. During a "show & tell", each team member presents the features developed in this release cycle, and the whole team discusses the features that had been delivered.

Table 1. Software used by the team (including remote worker)

Tool	Purpose	What the team used it for
Jira	Project planning / management/ticket tracking	Used as a dashboard for work coordination activities – In the sprint view mode (see Figure 1a) it provided a complete dashboard of the tickets to be completed, any blockers, and provided visibility of what the team members are working on. Also used as a repository for user stories and bug tracking
Bitbucket	Online Git repository	Used for code collaboration - pull requests, code review, and code commits. The code review comments appear inline against the code and allow interactive conversations about the code to take place online (see Figure 1b).
Hatjitsu	Online disposable poker rooms	Used for estimating and sizing user stories as part of the backlog grooming meetings. Fibonacci numbers are used for the planning poker sessions.
WebEx	On-demand communication, online meeting, web conferencing and video conferencing	Used for show & tell sessions per release that included all the project teams in the company and also sales teams.
HipChat	Chat and Instant Messaging (IM) application.	Used for IM and group chats within the project team. Also used for voice calling mainly by the office team.
SourceTree	Desktop client for the Git code repository	Used for code commits and maintenance along with comments on the nature of change – user stories, tickets, and bug fixes
MySQL	Database	Used as a backend for the PHP-based online system. Test environment of the database used extensively for debugging code with dummy data
Outlook	Email / Calendar	Used primarily for setting up tasks, meetings in the Outlook calendar - the meeting reminders were synchronised and thus all team members were aware of the various scheduled project-related activities.
Skype for Business	Voice calls / Group voice calls / Video calls / Group video calls	Used in integration with Outlook, mainly for video calls during meetings – mainly daily standups and retrospectives.
Appear.in	Video calls / Group video calls	Used for videoconferencing. Could be integrated with an HipChat IM chats and often used to declare impromptu requests for group video calls on the project IM channels
PHPStorm	Integrated Development Environment (IDE)	Used for writing, debugging, and testing PHP code. Often used in conjunction with SourceTree and Bitbucket.

The team made an effort to ensure that all team members, including the remote worker, participated in each of the Agile practices. For daily standups, the remote worker mostly participated via Skype and for other meetings, the remote worker joined via video call (either Appear.in or Skype). The remote worker travelled to the main office once every release. Referred to as 'touchpoints', these per-release in-person interactions were subject to workload and travel arrangements and sometimes did not happen.

English was the only language used for communication. All the team members were either native English speakers or spoke English fluently. All the team members (including the remote worker) worked in the same time zone.



(a) (b)

Figure 1. (a) Jira Sprint view (b) Bitbucket code review (Details deliberately obscured)

Data Gathering and Analysis

Initially, the researchers conducted observations and interviews in order to understand the organisation's context and working, and to identify a suitable focus for a more detailed investigation. Then a group discussion on remote working was held to which all members of the development teams were invited, including the remote workers.

Group Discussion

This discussion was structured as an Agile retrospective, i.e. participants were asked to consider the following four questions in turn: question 1 first, then 2, and so on:

1. What are we doing well?
2. What do we need to revise/rethink?
3. What should we be doing that we are not doing?
4. What are the blockers?

The session was conducted in one of the organisation's meeting rooms, involved eight developers including two full-time remote workers, and was facilitated jointly by one researcher and one Scrum Master. Each attendee was able to write their thoughts anonymously directly into the table using a shared wiki, and although the questions were considered one after the other, it was possible for participants to go back and modify their own earlier entries. After everyone had finished writing their entries, the group discussed each entry in turn. The session was audio recorded and the entries made against each of the questions were distributed afterwards and agreed. Subsequently the entries were themed.

There were 15 entries for question 1 (doing well). This included one-to-one conversations, small group meetings and a good selection of communication tools. There were 19 entries for question 2 (things to revise or rethink). This included communications for large group meetings, knowledge sharing, awareness, and whether or not to use remote pair programming. There were 12 entries for question 3 (what to do that we're not doing) and 14 entries for question 4 (blockers). Just under half of the answers to questions 3 and 4 (10 entries) related to infrastructure issues such as low bandwidth and noisy rooms; these were discounted from further analysis. Other entries related to pair programming, social

interaction, and awareness. We were particularly interested in questions 2 to 4 and themed the entries in order to determine a suitable focus for the next stage of research – see Table 2.

This analysis helped address RQ1, i.e. what kind of issues are faced in practice by a hybrid Agile team: infrastructure problems, pair programming, large group discussions, awareness, knowledge sharing and the difficulty of social interaction. Addressing RQs 2 and 3 required a more in-depth study of one team.

In-depth study of one team

This in-depth study used a combination of on-site observation, and shadowing of individuals and was completed over a three-week period coinciding with the start of a new release cycle.

The first author attended all of the key meetings during this period, i.e. daily standups (10), sprint planning (1), release planning (1), retrospectives (2), show & tells (1), planning /estimation meetings (1), and Scrum of Scrums (1). Other authors attended elements of the data gathering.

Table 2. Themes from the group retrospective on remote working and how they relate to discussion questions

Theme	Q2. What do we need to revise/rethink?	Q3. What should we be doing that we are not doing?	Q4. What are the blockers?
Pair programming	How/whether to attempt paired programming w/ remote workers Pair programming is rather cumbersome with remote worker	pair programming	
Large group discussions	very large group meetings Poor communications for remote workers with large group sessions		Communications big groups
Awareness	Multiple interruptions (other not aware you are busy / on a call) getting out of step with what is expected		presence awareness; being aware of the background "vibe"
Knowledge sharing/working together	planning work together Way of sharing ideas on the whiteboard		Fast response times for getting answers
Social interaction and familiarity	social interactions	"office life" webcam invite remote workers to virtual coffee breaks one communications tool but without losing out on office banter remote team building	interpersonal familiarity

In addition to observing the co-located office team, the first author spent a full day observing the remote worker at his home-office. Observation focused on interactions with colleagues in the office, interaction during meetings, use of software tools in relation to specific tasks, and use of the home office-space for day-to-day working.

The empirical data collected through these activities were observation notes, physical layout sketches, screen captures of the software artefacts used, still photographs, audio recording of the group discussion,

a document collating the views of those who attended the group discussion, and notes from two shadowing sessions.

Data Analysis

The data was analysed using the theoretical framework of distributed cognition (Hutchins 1995). Distributed cognition views collaborative work as one cognitive system, i.e. as being dispersed among individuals and between individuals and artefacts in the external environment (Halverson 2002). It focuses particularly on how information flows and is transformed within the system in order to achieve collaboration. Co-located Agile teams (Sharp and Robinson 2008) and dispersed Agile teams (Sharp et al 2012) have been analysed previously using this approach. The analysis here is based on distributed cognition in order to compare collaborative working in different settings. We draw on previous descriptions of distributed cognition, and a structured, empirical approach to it called DiCoT (Blandford and Furniss 2006) to analyse information flows and the use of artefacts within a hybrid team, specifically focusing on a remote worker and his co-located teammates.

DiCoT draws on ideas and representations from contextual design (Beyer and Holtzblatt 1997), together with a series of core distributed cognition principles. There are five themes in DiCoT: physical, information flow, artefact, evolution over time, and social structures. This analysis focuses on the first three themes because they are the most developed of the themes within software team analysis:

1. The *physical* theme which focuses on the physical environment within which the cognitive system operates, at whatever level of granularity is relevant, from the building or office layout to the positioning of items on a desk or noticeboard.
2. The *information flow* theme focuses on what and how information flows through the cognitive system, the media which facilitate that flow and how the information is transformed in the process.
3. The *artefact* theme focuses on the detail of artefacts that are created and used to perform the activity under study.

Blandford and Furniss (2006) identify 18 principles from distributed cognition which can be loosely categorised according to these three themes (see Table 3).

DiCoT Analysis

In this section, a comparative analysis of the co-located office team and the remote worker is presented using the DiCoT framework; this in-depth analysis addresses RQs2 and 3. Figure 2 shows the information flows and artefacts: square or rectangle shapes depict physical artefacts, virtual artefacts are depicted with circular or oval shapes, and the line separating the workers and the physical artefacts shows that the physical artefacts are not directly in the line of sight for the co-located office team.

Physical Layout

The physical layout of the co-located team's office and the remote worker's home office differed in terms of how the space was used and how the working environment supported the work activities. In addition, the remote worker's *horizon of observation* (see Figure 3) was limited to virtual artefacts that were visible only through the various tools being used, while the horizon for those in the office was richer, although possibly more distracting.

The following discussion provides a comparison of the co-located office team and the remote worker in relation to the principles of the Physical layout theme. Two principles, *perceptual* and *naturalness* (see Table 3), are not included since they were found to have limited relevance for this discussion.

Space and Cognition

The co-located office team used the physical space around them extensively. Various pieces of project-related information e.g. release schedules, statistics, and high priority defects from the customers were displayed on the walls. They used the physical scrumboard in the daily standups with post-it notes to record tickets, for discussions in various meetings; on their desks post-it notes were used to track individual activities. The use of a physical scrumboard and the significance of post-it locations on that

board has been noted before in co-located teams as a powerful artefact in team collaboration and co-ordination (Sharp and Robinson 2008). In contrast, the remote worker entirely relied on the virtual environment for his day-to-day work and did not use post-it notes or the available physical space to support his working. Although he used scraps of paper for minor tasks while debugging, he did not store those scraps for subsequent use.

Table 3. The DiCoT Framework

Physical Layout
Space and cognition: the use of space to support activity, e.g. laying out materials
Perceptual: how spatial representations aid computation
Naturalness: how closely the properties of the representation reflect those of what it represents
Subtle bodily supports: any bodily actions used to support activity, e.g. pointing
Situation awareness: how people are kept informed of what is going on, e.g. through what they can see, what they can hear and what is accessible to them.
Horizon of observation: what an individual can see or hear (this influences situation awareness)
Arrangement of equipment: how the physical arrangement of the environment affects access to information.
Information Flow
Information movement: the mechanisms used to move information around the cognitive system
Information transformation: when, how and why information is transformed as it flows through the cognitive system
Information hubs: central focuses where information flows meet and decisions are made.
Buffers: where information is held until it can be processed without causing disruption to ongoing activity.
Communication bandwidth: the richness of a communication channel
Informal and formal communication
Behavioural trigger factors: cause activity to happen without an overall plan
Artefacts
Mediating artefacts: used to perform the activity
Creating scaffolding: how people use their environment to support their tasks, e.g. creating reminders of where they are in a task
Representation-goal parity: how artefacts in the environment represent the relationship between the current state and goal state.
Coordination of resources: the resources (e.g. plans, goals, history and so on) that are coordinated to aid action and cognition.

Subtle Bodily Support

A number of physical gestures were used within the co-located office team including pointing to a screen by hand to discuss specific code or UX matters during a group or one-to-one interaction. During a one-to-one interaction the team members also used the cursor or mouse to point to the screen. Instead, the remote worker relied on screen sharing or screen swapping to discuss coding or testing issues. In consequence, the remote worker almost exclusively relied on the screen cursor or mouse to draw attention to a specific part of the screen.

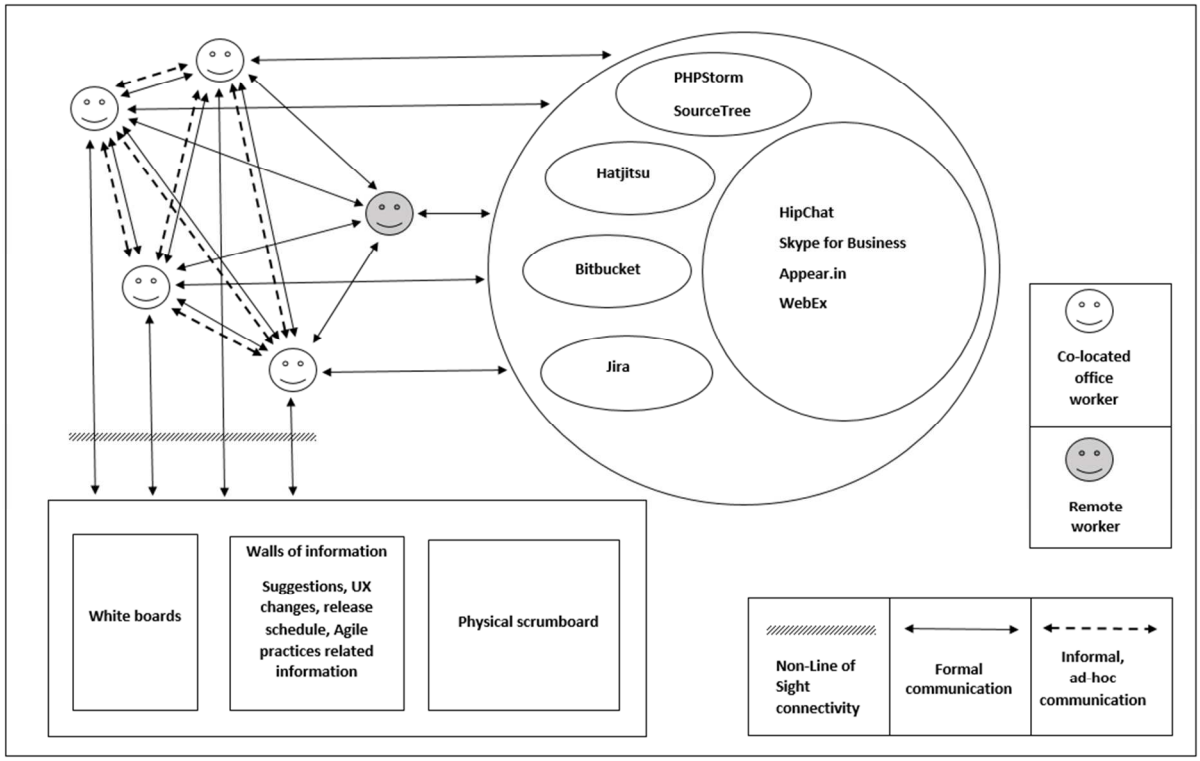


Figure 2. Information Flows and Artefacts

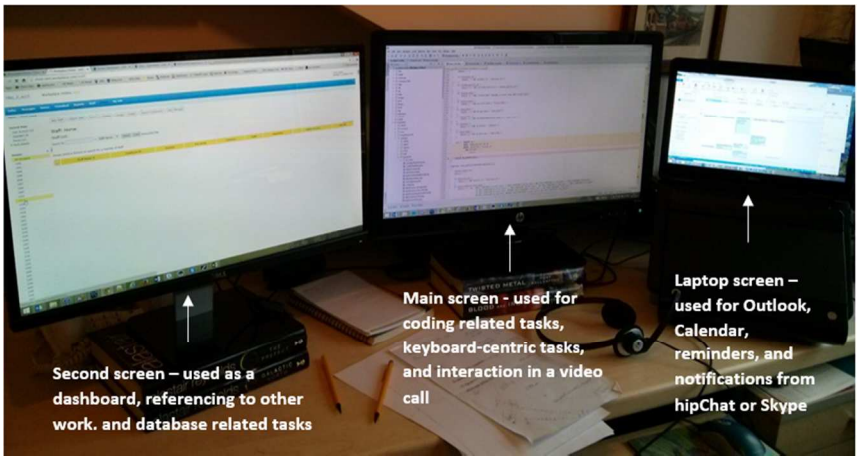


Figure 3. The remote worker's horizon of observation

Situation Awareness

The co-located office team members and the remote worker both relied on Outlook emails, Outlook calendar for meeting invitations and reminders, Jira scrumboards, and announcements on the HipChat instant messaging (IM) Channel as the primary means of being aware of various project-related activities and milestones. However, while the co-located office team members participated in-person in various standups and meetings, the remote worker only had a virtual presence. Thus the office team had an additional awareness about project-related activities and events built through offline conversations and interactions which the remote worker was not part of.

Horizon of Observation

The co-located office team members could see or hear a number of things: the informal office chatter, frequent conversations about the code, any bugs or issues in the environment, and impromptu discussions about software development, and coding practices. They also had access to other project related information including planned release details, development and test environment-related stats and details, and information on Agile practices (e.g. definition of DONE, 'Jumper' items) posted on the office walls. It must be noted though that although available, the physical artefacts (i.e. the physical scrumboard, the walls of information, and the white boards) were not directly in the line of sight of the co-located office team as shown in Figure 2.

The remote worker's horizon of observation was centred on the two screens and a laptop as shown in Figure 3, i.e. on the information available through his software tools. He was on a par with the co-located office team while participating in the discussions on the HipChat IM channel but was not part of the offline discussions that complemented this virtual discussion. The remote worker had no visibility of the physical scrumboard or other project-related information posted on the office walls.

Arrangement of Equipment

The co-located office team members and the remote worker each had access to a laptop and two large screens. Within the co-located office team, the arrangement of this equipment differed according to each individual's preference. Figure 3 shows how the remote worker preferred to arrange this equipment – with the laptop screen on the right used mainly for notifications, emails, and reminders. The remote worker used the two big screens for the bulk of the coding, testing, and database-related tasks. Although he had a printer, he used it only sparingly.

In addition to the computing equipment, the co-located office team members also used other equipment such as white boards, the walls of information, and the physical scrumboards – arrangement of this equipment was in mutual concert with the requirements of other project teams. Co-located team members occasionally used printers mostly for printing documents during a meeting.

Information Flow

For the co-located office team, the information flows took place online and offline – formally and informally. The information flows for the remote worker were limited to the virtual environment.

The following discussion does not cover the *behavioural trigger factors* principle (see Table 3) since no significant differences were identified between the co-located office team and the remote worker.

Information Movement

The co-located office team members used the physical scrumboard to relay project-related information along with Jira. On the physical scrumboard, the tickets were recorded on post-it notes that conveyed key pieces of information. Every ad-hoc and unplanned activity in relation to the project, use of software tools, and interaction between the co-located office team members contributed to information movement. The remote worker participated in information movement primarily through the software tools. He did not have any interaction with the physical scrumboard and interacted with the physical environment only when he was in the office as part of the regular touchpoints.

Information Transformation

Within the co-located office team, information was transformed in every instance of face-to-face interaction – whether planned meetings (e.g. standups, planning poker, or retrospectives), individual interaction, or ad-hoc conversations. Team members took part in the transformation in the virtual environment (e.g. Jira, HipChat, or Bitbucket) in addition to the offline interactions. In contrast, the remote worker had no access or participation in the office conversations. His participation in information transformation was entirely in the virtual environment mainly through Jira, Bitbucket, SourceTree, Skype calls, or HipChat IM channels.

Information Hubs

For the co-located office team, the main information hubs were Jira, the physical scrumboard, Bitbucket, SourceTree, and HipChat IM Channels. For them the other less important hubs were the white boards. For the remote worker the information hubs were exclusively virtual and thus did not include the physical scrumboard or the white boards. The co-located office team members preferred HipChat as an IM channel for online group conversations. However, it was less useful to the remote worker since it often contained formal communication and informal chat on the same channel which he could not easily distinguish.

Buffers

Information regarding various practices, UX changes, upcoming releases, specifics of Agile practices in a team (e.g. definition of DONE, 'Jumper' items), test and development environments, and new features was scattered throughout the team's office on various walls and white boards. These buffers were only available for the co-located office team. The remote worker had no access to these buffers.

Communication bandwidth

Within the co-located office team, a number of ad-hoc work-related conversations often took place as the team preferred a hands-on approach that de-emphasised excessive documentation in line with Scrum practices. Face-to-face interaction resulted in higher bandwidth communication. The frequent offline interactions offered ambient knowledge and tacit knowledge sharing. They preferred HipChat IM channel for sharing project information or project-related conversations with the entire team (including the remote worker).

For the remote worker all the communication was virtual – screen sharing with Skype to discuss coding or testing issues. In contrast to the offline interactions, these virtual interactions were premeditated, task-specific, and far less rich in the knowledge shared through them. Unlike the co-located office team members, the remote worker's participation in HipChat IM channels was not complemented by offline conversations on the same topics and thus he derived less value from such information exchanges.

Informal and Formal Communication

Within the co-located office team informal communication was routine and examples of it significantly outnumbered formal communication. Frequent ad-hoc conversations about code, fixing code, code related issues, or to brainstorm ideas were common amongst the co-located office team members. Some of the conversations about code originated from a discussion on Bitbucket and were continued offline. The co-located team members preferred to speak with someone in the office as it was more likely to result in a quick resolution. Thus they were likely to contact the remote worker only when the resolution of the task depended on him.

The remote worker and the co-located team preferred to contact each other or speak directly only when required. As a result, the remote worker's communication with the co-located team was almost entirely on a required basis specific to work-related tasks or activity. He took part in the HipChat IM channel occasionally and participated in code-related discussions mostly through Bitbucket.

Artefacts

A key project-related artefact used for tracking purposes was the ticket. A ticket is a reference to a requirement (in the form of a user story), a defect in test or live environment, or a new feature request directly communicated by a customer. Tickets were referenced in all project-related meetings and formed the bulk of the work done by the team as a whole, both co-located and remote. The ticket in Jira contained many details including progress information, any screenshots, related HipChat discussion, and a log of activity (including details of who had worked on it) as shown in Figure 4(a).

In contrast to the Jira ticket, the ticket used on the physical scrumboard was more concise. It was hand-written on a post-it note and contained key information for use at the time of the daily standups. An outline of the ticket as it was used in the physical scrumboard is shown in Figure 4(b).

The software tools, i.e. the virtual artefacts used by the co-located team and the remote worker, were the same and have already been listed in Table 1. A Project tracking system like Jira (see Figures 1a and 4) or a code review system like Bitbucket (see Figure 1b) that provided the same information to the remote worker (as to the office workers) were important artefacts when working effectively from a remote location. The use of software tools such as Jira, Bitbucket, Hatjitsu, or SourceTree did not differ for the remote worker and for those in the office. The differences, in relation to artefacts, between the co-located office team and the remote worker are covered below. Because the main focus was the software tools, and hence all team members used the same artefacts, two principles, *representation-goal parity* and *coordination of resources* (see Table 3), are not discussed.

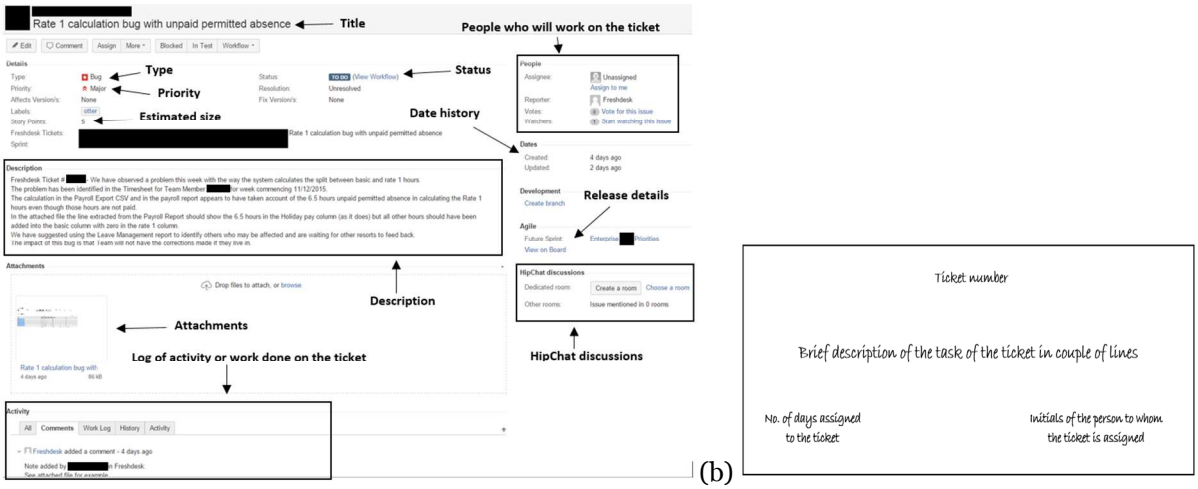


Figure 4. (a) Jira Ticket (Details deliberately obscured) (b) Physical scrumboard ticket

Mediating Artefacts

The co-located office team predominantly used virtual artefacts but also made use of other physical artefacts including the ticket as depicted in Figure 4(b). In the office, the team members favoured using HipChat for voice calls. They also used HipChat frequently to post information to a group and preferred it as its IM capabilities could be combined with Jira. Unlike the co-located office team, the remote worker preferred Skype over HipChat since it provided better audio reception and in his view, had a better notification system for missed messages or calls and volume of messages or calls.

Creating Scaffolding

The co-located office team members used post-it notes extensively as a means of making notes, creating reminders, keeping track of tasks in the form of tickets, and for brainstorming during team meetings. When used for their individual needs, the quantity, significance, and importance of the post-it notes differed based on each individual. The co-located office team also used white boards and the walls to convey information, highlight important deadlines, and invite opinion within the team.

The remote worker did not use post-it notes. He relied on the Outlook calendar to keep track of important reminders and tasks. In the absence of post-it notes, the remote worker made notes electronically – often in a Word document. These notes were meant for his own reference, not shared with the team. The importance of these notes varied; they contained details such as updates made on tickets in Jira, nature of changes made to the code, or comments on the code in SourceTree. The remote worker had very limited physical space available and did not use this space to aid his day-to-day work. In contrast to his office teammates he relied almost entirely on the virtual environment to support his work.

Discussion

Three research questions were asked at the beginning of this paper. RQ1 (what issues are faced by a hybrid Agile team) was addressed through a group discussion retrospective. The issues identified fall into five themes: pair programming, large group discussions, awareness, knowledge sharing/working together, and social interaction and familiarity.

RQ2 and RQ3 were both addressed by a distributed cognition (DiCOT) analysis of one hybrid team. This analysis suggests that collaboration in a hybrid Agile team is achieved through a combination of online support tools, individual interactions and group interactions (RQ2). Moreover, the remote worker's place in the collaboration differs from his co-located office workers in two distinct but related areas: the role of physical artefacts and the reliance on virtual artefacts and tools; and the level of engagement in information flows and information transformations (RQ3). While these findings are unsurprising on the one hand, they have particular significance for remote working in Agile software teams because of the need for tight collaboration that underpins the Agile philosophy. Moreover many of the tools listed in Table 1 support communication and coordination rather than continuous collaboration.

Artefacts and Information Flows

Virtual artefacts and their supporting tools are key *information hubs* for all team members. For the remote worker, these virtual artefacts dominate and shape their *situation awareness,* and they are the core of their *horizon of observation*. They are the primary representation of information for both remote worker and co-located team members, and the representations supported by these tools, i.e. the virtual artefacts themselves, are therefore important for collaboration. Co-located office workers supplement access to the virtual artefacts with different representations such as the less formal post-it note tickets and the physical scrumboard.

Co-located team members also supplement this information with *informal communication* and ad hoc discussion. Whereas the virtual artefacts were central to all the information flows to and from the remote worker, in the case of the office workers a number of indirect, informal *information flows* existed. For the co-located office team the extent of informal information flows outnumbered the formal information flows. The remote worker's participation in informal information flows was limited to face-to-face touch points and the HipChat IM channels. This meant that unlike the co-located office team all information exchanges could be effectively traced or tracked in the case of the remote worker.

The co-located office team members often solved coding, testing, and other software-related issues through ad-hoc, unscheduled conversations. Information including database or software environment breakdowns and any burning issues from live environments was available easily without those in the office making a deliberate, sustained effort to look for it. The remote worker, on the other hand, only interacted when a specific task or issue was brought to his attention.

For the co-located workers, the level of *information transformation* experienced is higher than that of the remote worker. The remote worker is not embedded in the informal support context that the co-located team members provide. In addition, the remote worker has very limited access to *informal communication* and ad-hoc discussions. This results in the *communication bandwidth* being lower for the remote worker than it is for the co-located office team.

All team members used the HipChat IM channel for informal communication. As HipChat was used for both social and work-related information, sometimes discussions were difficult to follow for the remote worker as he missed the office context. Some conversations could be overlooked by the remote worker missing some important information which was not translated into the virtual artefacts he focused on. Instead, the remote worker put effort into maximising his opportunities to build knowledge by tracking and retaining as much information through the virtual artefacts as possible.

The remote worker is not included in all *information flows* and is not engaged with all of the *information transformations* as his co-located teammates. In addition, more of the information flows around the remote worker are based on *formal communication* activities than those around the co-located team members. One consequence of more *formal communication* is that sharing information requires a

conscious decision on the part of those involved, and hence the remote worker is only included when a teammate deems it necessary, and so communication with remote workers was selective.

The above observations on artefacts and information flows are directly comparable to the conclusions from the DC analysis of a dispersed team conducted by Sharp et al (2012). In this case (where individual developers are all 'remote'), the virtual artefacts were also central *information hubs*, and *communication* was more formal and deliberate. However because the hybrid team is 'imbalanced', i.e. the majority (in this case four) of the team members are co-located and only one is remote, the effects of these issues are especially pronounced. In addition, these issues are particularly important as this is an Agile team for whom the need for unimpeded communication is essential.

Tools for Collaboration or Coordination

An important difference between the remote worker and the co-located office workers is how they relate to and employ the tool support.

Jira, the main source of information for coordinating and tracking project-related activities was central to the working of all the team members but its relevance and significance to the team members differed. To the remote worker, Jira created an equal opportunity to participate by providing the most current source of information about the tasks assigned to other team members, tickets pending, tickets in progress, release schedules, user stories, and estimation of story points amongst other data it provides. With Jira being the main and the most authoritative source of information, the remote worker did not 'miss out' as he would have if the physical scrumboard was the main source of information. The co-located office team combined Jira with other information available offline through interactions and discussions with colleagues but for 'factual' information they were reliant on Jira the same way as the remote worker.

When information is entered into any software repository, e.g. Jira, it has to be transformed in order to comply with the specific representation supported by the system. All team members have the opportunity to engage in this transformation to a degree, but for the co-located team members that opportunity differs because unlike the remote worker they engage in *informal communication* and have richer interpersonal interactions in team meetings. The remote worker sees the information in Jira but his participation in the *information transformation* that took place before it is entered in Jira is limited compared to his co-located teammates. Abdullah et al (2010) used communication theory to study the influence of interactions with work artefacts and other team members on the understanding of requirements (user stories) and consequent actions. They predicted that if one of the team members is absent, or if team members are distributed, then discoordinated actions may result. This indicates that the variable engagement by the remote worker and his co-located teammates may have consequences for the remote worker's understanding of requirements and consequent actions.

Bitbucket supported a different kind of interaction between team members that focused on code production, the team's central purpose. More than one of the developers we talked to revealed that the introduction of Bitbucket resulted in conversations about code, coding standards, and ensuring quality of work that previously did not happen. The senior team members including the remote worker suggested that unlike Bitbucket, the previous code review process functioned more as a checklist. The remote worker, in particular commented that using Bitbucket made him feel like a key stakeholder who had ownership of the software they were building as a team. This is relevant since he was providing feedback on others' work and also receiving suggestions about his work in an online, highly visible context. This fostered the feeling that he was contributing value to the team; team members could also hear his personal perspective in a different light. The interactive, highly collaborative nature of the online platform was thus the key to enabling the remote worker to function more cohesively with the co-located office team.

The transparency and visibility provided by a tool like Jira was crucial to fostering the kind of team ethos that is deemed crucial for Agile teams. Although the remote worker lost out on ambient knowledge, he had access to exactly the same project data and in consequence his capability was better integrated with the rest of his team. Jira showed a dashboard with project status, a snapshot of work in progress and any blockers posing a problem. Jira was used to support co-ordination, i.e. to make sure that team members knew what each other was doing and that they had a common goal. Bitbucket, on the other hand, was used collaboratively, i.e. the team members worked together more effectively to perform the task.

Bitbucket provided deep engagement with the code – as an owner and as a reviewer. Although code review existed before, Bitbucket cultivated the practice, caused additional conversations to take place (online and offline), and enabled a more meaningful collaboration with the code online. For the remote worker, it allowed his voice to be heard in relation to the main product, the code, and in effect strengthened his contribution to the team cause.

Reflections on the DiCoT framework

The DiCoT framework offers a rich structure with which to analyse collaborative working practices. It distills the key elements of distributed cognition into a practical framework, but it does not capture its full complexity. This is both a strength (because it is easier to apply) and a weakness (because it loses some of the full theory's subtleties). Although the themes of physical structure, information flows, and artefacts enable collaborative practices to be evaluated in a structured, systematic way, the challenge to the authors was to capture the human dynamics that made collaborative work in the form of remote working possible. The remote worker in this case had been with the organisation for a number of years and had been party to its transition from traditional waterfall-based development to scrum-based Agile working. His subject matter expertise extended beyond programming capability and included an understanding of organisational culture and a degree of resilience and adaptiveness that contributed to effective remote working. Similar observations can be made about the resilience of his co-located team-members that allowed the remote worker to collaborate effectively. These latter observations were not derived from the DiCoT analysis but directly from the fieldwork and the authors feel that this kind of organisational and individual adaptiveness is an aspect that the DiCoT framework does not fully capture.

Although the DiCoT framework includes themes focussing on 'evolution over time' and 'social structures', these are not sufficiently developed in relation to software team analysis. Whether exploring these themes would have made any additional contributions to this data analysis is difficult to determine. This suggests that there is a need to consider newer ways in which the themes in the DiCoT framework could capture the insights about the human dynamics that allow the collaborative working, remote working, or hybrid working to be effective and usefully productive

Threats to validity

Internal validity. Given that the data was primarily collected by the first author, a number of intensive discussions took place within the research team in order to limit the influence of individual opinion. The findings of this study (including the DiCoT analysis) have been presented and discussed with key members of the participating team and a member of the management to address potential critiques of representativeness and rigour.

Construct validity. The definition of what constitutes a remote worker depends on the company context, and there is limited literature on remote working in Agile teams. Existing literature on dispersed, distributed, and hybrid teams was examined to mitigate this threat.

Conclusion validity. The conclusions here are specific to the team's size, the software tools used, and the remote worker. Further research on hybrid teams with different collaborative tools, workers in different time zones, and non-native English speakers is needed to assess the validity of these conclusions.

External validity. This study focuses on the circumstances specific to one organisation, with a particular focus on one hybrid Agile team with a remote worker. This limits the extent to which the conclusions presented here can be generalized. However, factors that might impact on transferability such as the tools used, the tasks under consideration and the roles of the remote worker, are common in Agile software development teams. The tools used are standard Agile practice, the tasks undertaken are typical software engineering tasks and the role of the remote worker was that of a typical software developer.

This study can be seen as a single-case mechanism experiment (Wieringa, 2014). Generalisation can be achieved through analogy (analogic generalisation) and explanation (abductive inference). Analogical generalisation takes a situation that has similar characteristics to the one studied and discusses whether the circumstances are similar enough to allow the conclusions to be applied to that situation. In this case it would require the circumstances of a different team with one remote worker to be characterised and the applicability of these findings to be discussed. With abductive inference, an explanation of activity can

help support analogic generalisation. The explanations of the studied team, provided in previous sections, gives information to explore whether the findings can be applied to another team (Wieringa, 2014).

Conclusions

A hybrid team consists of a remote worker and his co-located teammates, and hence collaboration in this context has similarities with a dispersed team and with a co-located team. This is no surprise. However whereas in a dispersed context and in a co-located context, all team members have equal opportunity for collaborative activities, in a hybrid team, opportunities are imbalanced. This creates potential disadvantages as well as advantages for both co-located and remote workers.

For the remote worker, potential disadvantages include the well-known issues of being isolated and hence excluded from the knowledge network that his co-located teammates are embedded within. For the co-located workers, informal communication and additional information hubs may distract them from their main purpose. Potential advantages are the reverse of these: that remote workers can focus on the task in hand, and that co-located workers have a rich set of information available to them. Olson & Olson (2014) call this the *blind and invisible* stressor in distributed teams.

However you view these advantages and disadvantages, it is clear that collaborative platforms and communication mechanisms are crucial for integrating remote workers' capability into an otherwise co-located Agile team. More specifically, the extent to which the remote worker valued Bitbucket as a tool was striking. Bitbucket provided a higher degree of personal and professional validation to the remote worker – not because of the underlying activity (code review) but because of the interactive, social manner in which the activity could be conducted. Not only is the remote worker collaborating with team mates, but he is engaging meaningfully in a task that is central to the team's purpose: developing code. The co-located team received informal and tacit feedback for their work frequently, and so Bitbucket did not represent the same level of validation. Artefacts are often the means to an end. In this case Bitbucket just happens to be one of the available means of doing code review. However the effectiveness of the interface and the manner in which it reinforced the collaborative, communicative nature of a relatively mundane task is worth highlighting and requires further investigation.

Olson & Olson (2014) classify the tools to support distance work under: communication, coordination, information repositories, and computational infrastructure. In an Agile team most of these categories of tools are used as standard, and this study highlighted the principles that tools need to support. They need to create equal opportunities for engagement or isolation, particularly in terms of *situation awareness*, *horizon of observation*, *communication bandwidth* and the level of *information transformation* for all members of the team, including the remote worker. These findings may not appear to be surprising, but their implications for Agile software development are significant; Agile working relies on close and continuous collaboration in a way that other remote working situations do not necessarily require. Agile teams are attentive to their ways of working and are capable of adapting to new tools to better support their work. Our contribution highlights the importance that tools have in supporting *meaningful* engagement and *tight* collaboration in Agile remote working, including continuous knowledge sharing. For example, in our study the remote worker commented that the introduction of Bitbucket provided a higher degree of personal and professional validation to his work.

For researchers this work illustrates the working practices of hybrid Agile teams with remote workers, provides an insight into how a remote worker adapts to the constraints of not being in the office, and highlights the key differences between a co-located office team and the remote worker. Practitioners would benefit from considering both the technical and the social aspects in any hybrid situation.

The main driver for businesses to adopt remote working is the need for specific capability and subject matter expertise to be integrated into the team. This leads to the third key observation, which is that ensuring sufficient transparency and visibility of project-related activities through collaborative platforms is critical for the remote workers and it can also strengthen the effectiveness and cohesiveness of the co-located team (as seen with the use of HipChat or Bitbucket).

References

Abdullah, N. N. B., Sharp, H., and Honiden, S. 2010. Communication in context: a stimulus-response account of Agile team interactions. *Agile Processes in Software Engineering and Extreme Programming*. Springer Berlin Heidelberg, 166-171.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. 2001. *Manifesto for Agile Software Development* (February 2001). Retrieved November 12, 2015 from http://www.Agilemanifesto.org.

Beyer, H. and Holtzblatt, K. 1997. *Contextual Design: Defining Customer-Centered Systems*. Elsevier

Blandford, A. and Furniss. D. 2006. DiCoT: A Methodology for Applying Distributed Cognition to the Design of Teamworking Systems. *Interactive systems. Design, specification, and verification*. Springer. 26-38.

Halverson, C.A. 2002. Activity Theory and Distributed Cognition: Or What Does CSCW Need to DO with Theories? *Computer Supported Cooperative Work (CSCW)*. 11, 1-2, 243-267.

Holmström, H., Fitzgerald, B., Ågerfalk, P. J., and Conchúir, E. O. 2006. Agile practices reduce distance in global software development. *Information Systems Management*. 23, 3, 7–18.
Hossain, E., Babar, M. A., and Paik, H. 2009. Using Scrum in Global Software Development: A Systematic Literature Review. In *International Conference on Global Software Engineering*. IEEE.

Hummel, M., Rosenkranz, C., and Holten, R. 2013. The Role of Communication in Agile Systems Development. *Business & Information Systems Engineering*. 5, 5, 343-355.

Hutchins, E. 1995. *Cognition in the Wild*. MIT press.

Jalili, S. and Wohlin, C. 2011. Global Software Engineering and Agile Practices: A Systematic Review. *Journal of Software Maintenance and Evolution: Research and Practice*. DOI:10.1002/smr.561

Olson, J.S. & Olson, G.M., 2014. How to make distance work work. *Interactions*, 21, pp.28–35. Available at: http://dl.acm.org/citation.cfm?doid=2590181.2567788.

Paasivaara, M., Durasiewicz, S., and Lassenius, C. 2008. Using scrum in a globally distributed project: a case study. *Software Process: Improvement and Practice*. 13, 6, 527-544.

Rizvi, B., Bagheri, E. & Gasevic, D., 2015. A systematic review of distributed Agile software engineering. *Journal of Software: Evolution and Process*, 27, pp.723–762.

Sharp, H., Giuffrida, R., and Melnik, G. 2012. Information Flow within a Dispersed Agile Team: A Distributed Cognition Perspective. *Agile Processes in Software Engineering and Extreme Programming*. Springer, 62-76.

Sharp, H., Robinson, H., Segal, J., and Furniss, D. 2006. The Role of Story Cards and the Wall in XP teams: a distributed cognition perspective. In *Proceedings of the conference on Agile 2006*, 65-75. IEEE Computer Society Press, Minneapolis
Sharp, H. and Robinson, H. 2008. Collaboration and Co-ordination in Mature eXtreme Programming Teams. *International Journal of Human-Computer Studies*. 66, 7, 506-518.

Staples, D.S. and Webster, J. 2008. Exploring the Effects of Trust, Task Interdependence and Virtualness on Knowledge Sharing in Teams. *Information Systems Journal*. 18, 6, 617-640.

Wieringa, R., 2014. Empirical research methods for technology validation: Scaling up to practice. *Journal of Systems and Software*, 95, pp.19–31.