

Communication Support with Scrum in Offshore-Outsourced Software Development

HELENE HJELMVIK



**KTH Computer Science
and Communication**

Master of Science Thesis
Stockholm, Sweden 2008

Communication Support with Scrum in Offshore-Outsourced Software Development

H E L E N E H J E L M V I K

Master's Thesis in Human Computer Interaction (30 ECTS credits)
at the School of Media Technology
Royal Institute of Technology year 2008
Supervisor at CSC was Ann Lantz
Examiner was Yngve Sundblad

TRITA-CSC-E 2008:117
ISRN-KTH/CSC/E--08/117--SE
ISSN-1653-5715

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.csc.kth.se

ABSTRACT

Over the passed decade Offshore–Outsourced Software Development to India has increased and continues trending upward. Companies have approached this new, distributed situation in several different ways. During 2007 the Swedish–Indian company Indpro AB introduced the agile method Scrum to their offshore development projects in Bangalore, India. Using agile methods demands a tight collaboration, which can be hard to achieve when the development is spread out.

The aim of this thesis was to study this environment by seven different qualitative methods over a period of six months, and it resulted in a description of the communication flow in the Bangalore office. The result reveals a lack of informal information exchange between the different actors in the cross–location projects and suggests how smaller infrastructural changes could enhance the computer–mediated communication in this environment. The discussion suggests using a more stringent application of Scrum in the future by creating and forcing more communication rules on the collaborators. By planning and executing regular formal and informal meetings a common ground can be founded, which will lead to an improved collaboration situation.

Kommunikationsstöd genom Scrum i mjukvaruprojekt outsourcade till Indien

SAMMANFATTNING

Under det senaste decenniet har mjukvaruutveckling outsourcad till Indien ökat, och det finns en klar tendens att denna utveckling kommer att fortsätta. Företag har angripit denna nya distribuerade situation på olika sätt. Under 2007 började det svenska indiska företaget, Indpro, använda den agila utvecklingsmetoden Scrum i sina utvecklingsprojekt i Bangalore, Indien. Att använda en agil metod kräver ett nära samarbete mellan de olika aktörerna inom projektet vilket kan vara svårt att uppnå om projektet är utspritt.

Syftet med denna studie var att studera denna specifika miljö genom sju olika kvalitativa metoder under en sexmånadersperiod. Studien resulterade i en beskrivning av kommunikationsflödet på kontoret i Bangalore. Resultatet avslöjar att det saknas stöd för informell information mellan de olika aktörerna i de utspridda projekten och föreslår hur mindre uppgraderingar av kontorets infrastruktur kan förhöja den datorstödda kommunikationen. Diskussionen rekommenderar även att en mer stringent användning av Scrum i fortsättningen genom att tvinga projektmedlemmarna att använda fler kommunikationsregler. Genom att planera och genomföra regelbundna formella och informella möten kan en gemensam grund skapas, vilket leder till ett bättre och mer gynnsamt samarbete.

FOREWORDS

Many thanks to the Indpro crew for good support and a great friendship and thank you Ann Lantz at KTH for a great mentorship.

KEYWORDS

Communication, Offshore, Outsourcing, Agile Development, Scrum, India, Sweden, Software development, Indpro AB, Communication, CSCW, HCI, Bangalore.

INDEX

1 Introduction	1
1.1 Problem	1
1.2 Aim	2
1.3 Delimitations	2
2 Background	3
2.1 Offshore software development	3
2.1.1 India and Bangalore	4
2.1.2 Collaboration between India and Sweden	4
2.2 Indpro AB	5
2.3 Terminology and definitions	6
3 Theory	9
3.1 CSCW	9
3.1.1 Communication over distance	10
3.1.2 Grounding in communication	11
3.1.3 Awareness in groupware	12
3.2 Agile software development	13
3.2.1 Scrum	14
3.2.1.1 Roles	14
3.2.1.2 Documentation	16
3.2.1.3 Meetings	16
3.2.1.4 Scrum over distance	18
3.3 Previous studies of offshore software development and agile methods	18
3.3.1 Background	19
3.3.2 Distributed offshore development	19
3.3.3 Communication	20
4 Method	23
4.1 Interviews	24
4.1.1 Structured interviews	24
4.1.2 Semi-structured interview	24
4.2 Observations	25
4.2.1 Structured observations	25
4.2.2 Ethnographically inspired observations	26
4.3 Participatory study	26
4.3.1 Participatory observations	27
4.3.2 Field experiments	27
4.4 Heuristic evaluations	28
5 Results	29
5.1 Bangalore office	29
5.1.1 Software engineers	29
5.1.2 Team Leaders	29

5.1.3 Projects	29
5.2 Scrum implementation	30
5.2.1 Documentation	30
5.2.2 Meetings	31
5.2.3 Roles	32
5.2.4 Scrum in practice	32
5.3 Synchronous Communication	35
5.3.1 Audio Meetings	35
5.3.1.1 Technical issues	35
5.3.1.2 User related issues	36
5.3.2 Instant messaging	38
5.4 Asynchronous Communication	38
5.4.1 Collaborative Workspace	39
5.4.1.1 Product and Sprint Backlog	40
5.4.1.2 Burndown chart	41
5.4.1.3 Team Discussion	42
5.4.1.4 Bug list	42
5.4.1.5 Shared documents	43
5.4.2 Email	43
5.5 Future collaboration	43
5.5.1 Shared mailing list	44
5.5.2 Meetings with all actors	45
5.5.3 All documentation on shared workspace	45
6 Discussion	47
6.1 Results	47
6.1.1 Communication and Scrum	47
6.1.2 CMC Usage	49
6.2 Method	51
6.3 Theory	52
6.4 Future work	53
7 Conclusion	54
7.1 Scrum Usage	54
7.2 Awareness Support	54
7.3 Hardware improvements	55
7.4 Future work	55
References	56
Appendix	59

1 Introduction

Over the passed decade offshore-outsourced software development has increased. To offshore is to relocate a business process to another country. Bigger companies often choose to open their own offshore unit in another part of the world. Small- and middle-sized companies might not have this option due to financial or resource related issues. The option is then to relocate the software development to an external company in another country. This phenomenon is called Offshore-Outsourcing. Indpro AB (Intelligent Development and Programming AB) gives Swedish companies another, unique opportunity: they provide offshore software development resources. The clients are given the opportunity to hire one or more programmer consultants in India and pay them by the hour.

Recently Indpro started working with an agile software developing method, one that demands a close collaboration between the Swedish client and the Indian developers. With today's advanced techniques, such distant collaboration is possible. It can even be executed in several different ways and Indpro has selected a method and different computer mediated communication tools to make it work. This thesis will study this specific environment and map the communication flow in and around Indpro's Bangalore office.

1.1 Problem

During 2007 Indpro has been implementing the agile method Scrum to their eleven active programming projects. The projects vary in size, character and to what level Scrum have been implemented. The common denominator of the projects is that the client is Swedish and the development is committed offshore in Bangalore, India.

Scrum is an agile method that demands that developers, stakeholders and project manager engage in a close collaboration. This collaboration could be suffering when the development is spread out. The vision at Indpro is to not to sell a complete solution, but simply to rent out the manpower. The idea is that

the client not only should act as stakeholders, but also lead the projects (i.e. supply projects with a project leader). This way of working, with a Swedish project leader and product owners located in Sweden and the programming team in India, put higher pressure on the communication between the two countries. There are several additional factors to consider in order to make this collaboration work. Among many are cultural differences, time zones and language barriers.

The task given was to study the communication flow in the organization and how the method Scrum, as a framework, can facilitate for the different actors. The result should suggest of how mediated communication can and should be used to get the best communication and collaboration support between Bangalore and Sweden.

The research questions in this thesis are:

1. What communication support do the agile methodology Scrum provide for an offshore–outsourced environment?
2. Describe the communication within the offshore–outsourced projects.
3. What collaboration technology is needed to enhance the communication and bandwidth¹?

1.2 Aim

The master thesis is a study at Indpro's office in Bangalore, India. The aim is to study the communication within the different projects and to understand what support the method Scrum provides. The result should suggest a method to enhance the communication bandwidth within Scrum, but also use of computer mediated communication tools for this specific offshore and distributed environment.

1.3 Delimitations

This study is analyzing a specific situation and give suggestions on how to improve it. It does not include implementation of the ideas nor to give a

¹ In this thesis bandwidth refers to communication channels used, as specified in chapter 3.1.1

complete analysis and solution to the problem. The research is describing one specific environment at Indpro under a specific state of implementation of the methodology Scrum. This is not a generalized solution to all similar environments. However the study, or parts of the study, can be used as a background and foundation when starting up distributed offshore-outsourced development.

2 Background

This chapter provides relevant background information on offshore software development in India and about the company Indpro AB. Section 2.3 is a wordlist that contains important words and phrases used further in the report. The list should work as a definition dictionary while reading the rest of the report.

2.1 Offshore software development

To offshore development is to move parts or the entire software development to another country. When a third-party company is hired offshore to execute the development it is called offshore outsourcing. Up until now offshore outsourcing have been very limited for IT companies, only 14 per cent of the Swedish IT companies outsourced IT during 2007 (Overby, Wikberg, Heymowska & Röhne, 2008). But through the globalization of the world economy the interest is growing. Popular offshore locations for Swedish companies are eastern European countries and Asia. Today India is the dominating offshore country worldwide. The advantages with India are that they have world-class engineering schools, which graduates more than two million resources a year. As of today India has the worlds largest number of software engineers (Swedish Trade, 2008). In India English has been retained as a language for communication and education, hence it is widely spoken and written by the Indian software engineers. Today 23 per cent of India's total IT-export is to Europe and 61 per cent to North America. The trend reveals that the share to Europe will rise in the near future. Swedish IT companies have already chosen to locate parts of their software development

in India, but Sweden does still have many unused possibilities in India (Eriksson Fogh, 2005).

2.1.1 India and Bangalore

In 1947 India got their independence from the British colonists. India has a rich cultural legacy and although they have been influenced by the western world they still have their own identity (Andersson, 2006). India kept the British educational system and English as one of two official languages. India has more than a thousand million in population and the country is split into 28 states. The diversity of the people is great: they speak different languages² and pray to different gods³. Despite this they have a strong cultural core that unite the population. Through a strong and growing middle class India is ranked as the fourth strongest economy in the world today (Ministry for Foreign Affairs, 2008)

Bangalore has a population of about ten million people and is located in the state of Karnataka and the common language is Kannada. It is a city with opportunities and therefore many are moving in from all over India and the city is growing each day. Bangalore is a modern city, a melting pot of all Indian culture. Today there is neither a dominating ethnical group nor a major language in Bangalore. It is called India's IT capital (Eriksson Fogh, 2005) and described as one of India's most modern and sexually equal cities (Andersson, 2006). Also Bangalore has numerous of non-Indian residents, due to all offshore industries located there.

2.1.2 Collaboration between India and Sweden

Yet over 100 Swedish companies are established in India (Swedish Trade, 2008). In the IT sector there are examples such as Ericsson, Volvo and Telelogic. Also several great Indian companies have offices in Sweden, such as Tata Consultancy Services, Wipro and InfoSys (Eriksson Fogh, 2005).

² India has about 15 major languages and 844 different dialects (Indian Embassy, 2008).

³ Hinduism is the major religion in India (~80%), second largest is Islam (~13%) and third is Christianity (~4%). Within hindusim there are several different sub-denominations. Also the division between the religions vary between the states (Indian Embassy, 2008).

Many cultural differences can be found between the two countries, both in population and geography. Climate, religion and social structure are some of the variables to respect when comparing the countries inhabitants. If one person from each country would execute a simple task such as defining the color of blue we would most likely get two different answers. When connecting two countries by an online tool there is no culture to follow, i.e. there are no cultural right and wrong. Both parties feel as if they are in their culture, hence a lot of misunderstandings can occur (Olson & Olson, 2000).

The time difference between India and Sweden is four and a half hour during winter and three and a half hour in the summer. This leaves a time box of several office hours that overlap and synchronous communication is possible between offshore and onshore locations. However the time zones can affect the communication through cutting down the availability (Olson & Olson, 2000). In Sweden office hours generally starts at eight o'clock in the morning. In India offices generally open at nine, which adds one extra hour to the joint time.

Language is also an obstacle in the collaboration between Sweden and India. Neither of the actors generally have English as the native tongue which can lead to misunderstandings.

2.2 Indpro AB

Indpro was founded and opened their first office in Bangalore, India in 2005. In November 2007 they have business in two different branches, software and CAD development. They have an office in Bangalore and their headquarter in Stockholm, Sweden. The biggest branch is the software development and that is the division that has been analyzed and studied in this thesis. When Indpro is mentioned further in this report, it only concerns this part of Indpro's business areas. Their business plan is to offer Indian resource solutions to Swedish clients. Within the company they have local knowledge and "know-how" in both Sweden and India, due to their

multinational background. Indpro has a Swedish management on location in Bangalore at all times.

Indpro has been growing during the passed years and the office in Bangalore change rapidly. During the study the conditions has changed several times, such as procedures, equipment and infrastructure available. This report is based on the state the organization and projects was in the beginning of the studies, which is further specified in the result chapter (see 5.1).

In the Bangalore office the Indian developers work close together with the collocated Swedish team leaders and the remote client. Indpro encourage the client to travel to the office in Bangalore in the start of the projects to meet the resources and acquire a cultural exchange. For the same reason Indpro value to always have Swedes at the offshore location.

2.3 Terminology and definitions

Agile development – A group of methods for software development.

Asynchronous communication – Non-simultaneous communication between two or more actors.

Awareness – The feeling of be conscious of other people in the shared environment, physical or digital.

Bandwidth (in Computer Science) – A measurement of how many bits per second can be transferred over a network.

Bandwidth (in CSCW) – A measurement of how much information that can be communicated using different computer-based tools.

Computer Mediated Communication (CMC) – Communication using digital tools to transfer the messages.

CMC – See Computer Mediated Communication

Computer Supported Collaborative Work (CSCW) – An academic field studying computer-based tools used for collaboration.

CSCW – See Computer Supported Collaborative Work

Distributed development – Development project processes are spread out across different worksites or locations.

Extreme Programming (XP) – An agile method based on programming in pairs.

Instant Messaging (IM) – Text based synchronous conversations over Internet.

IM – See Instant messaging

Locked – A term in Scrum used when a sprint is populated with content which is not editable anymore.

Offshore – Offshoring is defined as the movement of a business process done at a company in one country to the same or another company in another, different country.

Offshore Outsourcing – When company is subcontracting a business unit to a different company in another country would be both outsourcing and offshoring.

Outsourcing – Outsourcing is the movement of internal business processes to an external company.

Product Backlog – The base of a project using Scrum methodology. Filled with to-do items each one to four days of work for the development team.

Product Backlog Items – User stories that needs to be done in order to complete the project, together they form the Product Backlog.

Product owner – The role as stakeholder of the project using Scrum.

Scrum – An agile method for software development with characteristics such as close collaboration and short iterations.

Scrum master – Person responsible for that the team and product owner is sticking with the Scrum methodology.

Sharepoint – An online Microsoft tool for coordinating projects.

Skype – A digital tool with voice and IM messaging over Internet.

Sprint – Section of the product backlog (several items) that will take the development team one to four weeks to complete.

Sprint burndown chart – A diagram showing the relationship between cost and worked hours in a sprint.

Sprint demonstration/Demo meeting – Meeting after a sprint where the team will demonstrate what they have done for the product owner.

Sprint planning meeting – Meeting where all parties of the project gather

and plan for the upcoming sprint.

Sprint retrospective meeting – Meeting after a sprint between scrum master and team with the purpose to evaluate the sprint and give feedback to the team.

Sprint review meeting – Meeting after a sprint between scrum master and team with the purpose to evaluate the sprint and give feedback to the team.

Synchronous communication – Communication with delayed reply and feedback from other actors.

Test driven development – An agile method with focus on testing.

Wiki – A collection information web pages that is accessible and editable by all users.

XP – See Extreme Programming.

3 Theory

This chapter gives a background to the different fields relevant to this study. It presents a theoretical introduction to the field Computer Supported Collaborative Work (CSCW) to give essential background information on communication and collaboration theory in a computer mediated setting. Later is a basic presentation of the agile software development method Scrum. Further there is a section describing previously done studies on distributed and offshore-outsourcing software development using agile methods. Section 3 in the background chapter (2) contains a list of words and definitions related to this study.

3.1 CSCW

Computer supported cooperative work (CSCW) is a term that addresses how computer systems can support collaboration. It is a cross-disciplinary academic field that combines the technical design of groupware (i.e. computer based tools for collaboration) with a psychological, social and organizational understanding of the collaborative situation.

CSCW defines two dimensions of cooperation: time and space. Figure 3.1 is a matrix dividing the CSCW field into four areas to define the conditions for the collaboration.

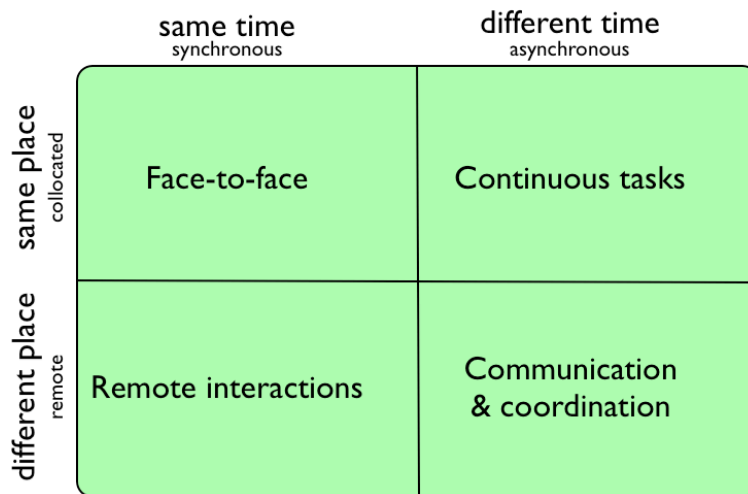


Figure 3.1: *CSCW Matrix.* Matrix demonstrating the dimensions of the CSCW field.

The matrix distinguishes between collaboration done synchronous and asynchronous, collocated and remote. Synchronous communication or collaboration take place in real time, this type of conversation can be done by voice or text. Examples of computer mediated communication tools that support synchronous collaboration are instant messaging (IM), chat rooms and videoconferencing. As for asynchronous communication is when the communication take place at different times, it doesn't rely on time dependent turn taking (Preece, Rogers and Sharp, 2002). Examples of digital tools that support asynchronous communication are e-mail, forum threads and wikis. In collocated work the collaborating actors share the same physical environment and remote signifies when the actors are separated geographically (Olson & Olson, 2000). Remote collaboration can be done in a numerous ways and the communication support is vital.

3.1.1 Communication over distance

When working together over a distance, the conditions for communication are different than if we share the same physical environment. Olson & Olson (2000) suggest one great difference between collocated communication and communication over distance is the spatiality. Human communication is built into the space it occurs. Hence if the common space is removed many important communication channels are closed. Subtle information and the context are an example of losses. They also argue that despite this, a new space is created and through this new possibilities are created where both success and failure can be observed.

The bottom line is that the more common ground that can be established the better communication and the higher the bandwidth is the easier you create common ground (Olson & Olson, 2000). Bandwidth in the CSCW field is different from the definition in the computer science field. Instead of counting how many bits per second that can be transferred over a network, it defines how much can be communicated. It measures how many channels are used to communicating a message to the receiver, examples of channels

are voice-, text- and video links (Preece et al., 2002).

To communicate and interact over a distance different technical tools are used. There are various kinds of tools on the market, using different communication channels and support different kinds of communication. Such technologies are commonly referred to as computer mediated communication or CMC (Preece et al., 2002). Successful communication does not only depend on the CMC used, but also on the sender and receiver of the messages. It is important to have an understanding between the different actors: a common ground.

3.1.2 Grounding in communication

When creating a common ground in remote communication there are several important obstacles to consider. Early Clark & Brennan (1991) discussed that it is easier to ground conversations when it is held face-to-face. Using a communication medium strain a higher rate of collaborative effort and in order to achieve successful communication as little effort as possible should be demanded. One of the obstacles in using communication mediums is that the communication loses its ground. The authors define grounding as the feedback from the receptor of the message that the message has been received. They defines eight constrains that different medium might impose.

1. Co-presence: The communicators are in the same physical place.
2. Visibility: The communicators can se each other.
3. Audibility: The communicators can hear each other.
4. Co-temporality: There is no delay in creating and receiving messages.
5. Simultaneity: All communicators can send and receive messages simultaneously.
6. Sequentiality: All conversations are exclusive.
7. Reviewability: The conversation can be reviewed.
8. Revisability: The communicators can alter each other's messages.

Although they claim that cost in one medium can also be traded off as its advantage if the medium and the communication purpose are coordinated.

The last two constrains are for example not possible in face-to-face communication. Hence in some situations face-to-face communication is not the best option.

A wider concept in the field is common ground, which implies that the communicators have a foundation in common on which they build their conversation. Olson & Olson (2000) include common ground as one of four key concepts of communications future, along with Coupling, Collaboration Readiness and Collaboration Technology Readiness. Further they develop the definition of common ground to include opinions of appearance and other understated information. Common ground can be created from any information about the other communicators that is available.

If the communicators have a common purpose and a common base the grounding in communication is easier. One example is that more subtle information can be fetched if the actors have a common ground (Clark & Brennan, 1991). A way to help creating a common ground is to use CMC with support for awareness.

3.1.3 Awareness in groupware

To be aware of others is simple when they are sharing the same physical space. When the workspace is in a computer system the awareness is harder to obtain, but to stay aware of the other actor is very important in such environment (Gutwin & Greenberg, 2002). It is not only the awareness of that there is other people there, but also to know what they are doing and how they are feeling. Olson and Olson (2000) describes that to know others mental state is vital when creating awareness. Awareness should give answer to the questions: Who, What and Where.

Awareness is described as the knowledge of an environment in real time. The awareness should be obtained through interaction and should not be the main objective in a task. Gutwin and Greenberg (2002) put the term awareness in a workplace context. They define workspace awareness as

awareness with focus shifted from the people to the workspace, or rather people's interaction with the workspace. Several aspects of a situation should be considered when designing a groupware to support awareness. They listed four boundaries to consider when approaching this problem:

1. The systems limitations of the groupware.
2. The environment of the shared workspace.
3. The task performed by the actors and the groups.
4. The characteristics of the groups that are using the system.

The workspace awareness is completely focused on the task the team is carrying out and the goal is to obtain useful data. This awareness is important because it helps the receiver to interpret the message in the correct way. Including direct references, demonstrations, manifesting actions and visual evidence. Successful workspace awareness can compensate for a missing common ground. This type of awareness is easier to create since it doesn't have to be as complete as the situated awareness.

3.2 Agile software development

There are numerous of ways of how to organize and work in software development projects. One approach is to use an agile method, which provides a conceptual framework for all stages of the development. The Agile Manifesto (see Figure 3.2) defines the principles behind agile development and is built up of four statements of how to approach software development. It highlights the importance of collaboration, interaction and a dynamic way of working. It sees to have all actors with beneficiaries in the project to be actively involved in the development planning and progress.

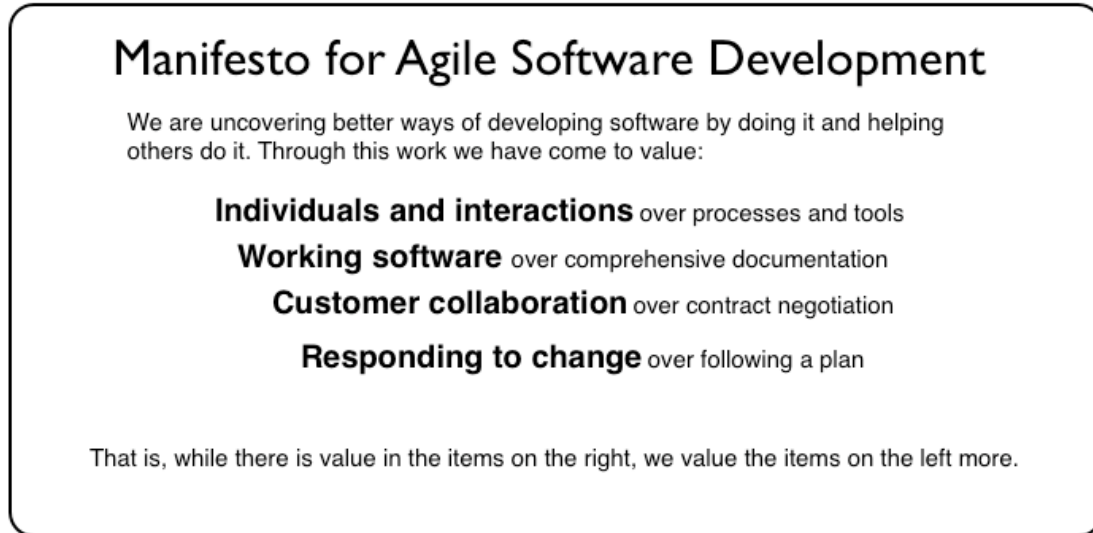


Figure 3.2. *The Agile Manifesto.* Demonstration of the most significant values of agile development (the Agile Manifesto, 2008).

There are many flexible methods born before and after the agile framework evolved in the mid-1990s. Examples of such are Scrum, Extreme programming (XP) and Test-Driven development (TDD). In the next sections Scrum will be explained more in detail.

3.2.1 Scrum

Scrum is an agile method that has grown in popularity over the passed years. It has a set of guidelines, but no strict rules, all according to the agile way of development. Scrum is named after rugby strategies due to its use of teams in the fight to possess the ball (Schwaber & Beedle, 2001). More than Scrum teams the methodology defines a number of significant other roles, documents and meetings. These will be further explained in the following sections.

3.2.1.1 Roles

Scrum divided the involved actors into three different roles (Schwaber, 2004):

Product owner – A person who represent the stakeholders and all people with interest in the final system. The product owner is responsible of the

specifications, requirements, return of investment and release plans, and also charge of the Product Backlog (see 3.2.1.2).

Team – The team are the programmers responsible for the actual software development. The team is self-managing and should come up with how to break down the product backlog into workable tasks. The team is responsible to reach the sprint goals.

Scrum master – A person who is responsible for that the Scrum process is followed correctly and in charge of all Scrum related questions. The Scrum master is also the link between the team and the management.

Scrum demands a close collaboration between the team members. Interaction, collaboration and individuals are highly valued in agile development (the Agile Manifesto, 2008)The Scrum methodology implements a framework for meetings and documents, they are agile but guidelines are set up for them. The Scrum process is iterative and can vary much between different projects. It gives a framework for what meetings should be held, who should be involved and how frequent they should occur. How the different documents and meetings are integrated in the iterative process is illustrated by figure 3.3 below.

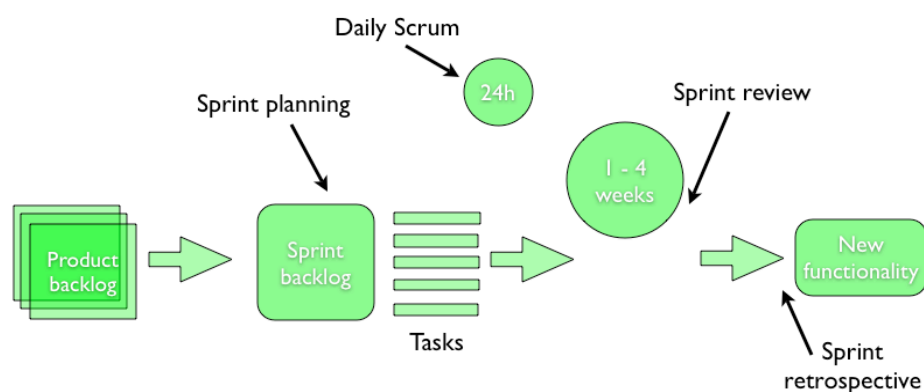


Figure 3.3: *Scrum Work Process.* The figure explains the iterative Scrum process with its meetings and documentation.

3.2.1.2 Documentation

When a project starts the first thing that happens is that the product owner set up a *product backlog* for the project. The product backlog is a set of business and functionality stories called *product backlog items* or *issues* are programming stories that needs to be done in order to finalizing the project. This list is dynamic and items and issues can be added by the product owner throughout the project (Schwaber & Beedle, 2001). Issues represent items that still need resolution before they are workable items. The items are time estimated by the team (Kniberg, 2007), or by the product owner (Schwaber & Beedle, 2001) this is done in how many programming days it will take for a pre-decided team size to finish the story. The product owner is also in charge of arranging the items according to their priorities in the project. The following information fields are recommended for each product backlog item (Kniberg, 2007): ID, Name, Importance, Initial estimate, How to demo and Notes.

The *sprint backlog* is another list of shorter programming stories, *tasks*. The tasks are to be estimated in hours for one programmer, the estimation can be done without the product owner in the second planning meeting explained in section 3.2.1.3. After a sprint is *locked* changes should not be made in the sprint, the team should be free to work with and prioritize the tasks as it like (Schwaber & Beedle, 2001).

3.2.1.3 Meetings

Common is that the team has a *daily meeting* where the team, lead by the Scrum master, meet and discuss yesterdays progress, today's tasks and issues that might have been raised since last meeting (Schwaber & Beedle, 2001). This meeting is also mentioned as a 15 minutes daily stand-up meeting where the team members gather in front of a whiteboard (Kniberg, 2007).

The product backlog is divided and locked into smaller iterations called *sprints*. The length of the sprints are set for each project and ranging from one and four weeks. What to work on in a sprint is decided in two *sprint*

planning meetings: One between the team, Scrum master and the product owner and another with the team alone (Schwaber & Beedle, 2001). The aim of the first meeting is to figure out what to include in the next sprint, ensure all requirements are clear and to set up a *sprint goal*. It is important that all actors are attending when deciding what items to *push* in the sprint in order to reach the sprint goal. The second meeting is for the team to organize the sprint work practically and time-estimate the work. It is also the time to set up the *sprint backlog*, decide on a date for a *sprint demo* and set a time and place for the daily scrum meeting (Kniberg, 2007).

When closing a sprint a *sprint demonstration (demo) meeting* (Kniberg, 2007) or *sprint review meeting* (Schwaber & Beedle, 2001) should be held. The sprint demo is held between the team, the Scrum master and the product owner. The aim of this meeting is for the team to demonstrate for the product owner, other stakeholders and other collaborating teams what they have finished during the sprint. This meeting can be done in several different ways, but should be a social event where different actors can interact and discuss their work from their different perspectives (Kniberg, 2007). The Scrum master is the moderator of this meeting and should meet with the team in beforehand to prepare the meeting (Schwaber & Beedle, 2001).

After the sprint is closed a *sprint retrospective meeting* is held. This is a time to review the sprint itself and the process and all actors should attend. This is the time to evaluate the estimations and compare them with the actual velocity of the work. It is then helpful to draw a *sprint burndown chart*. A chart showing the sprint cost related to consumed time. This can also be a good time to share information between different team in a distributed environment (Kniberg, 2007).

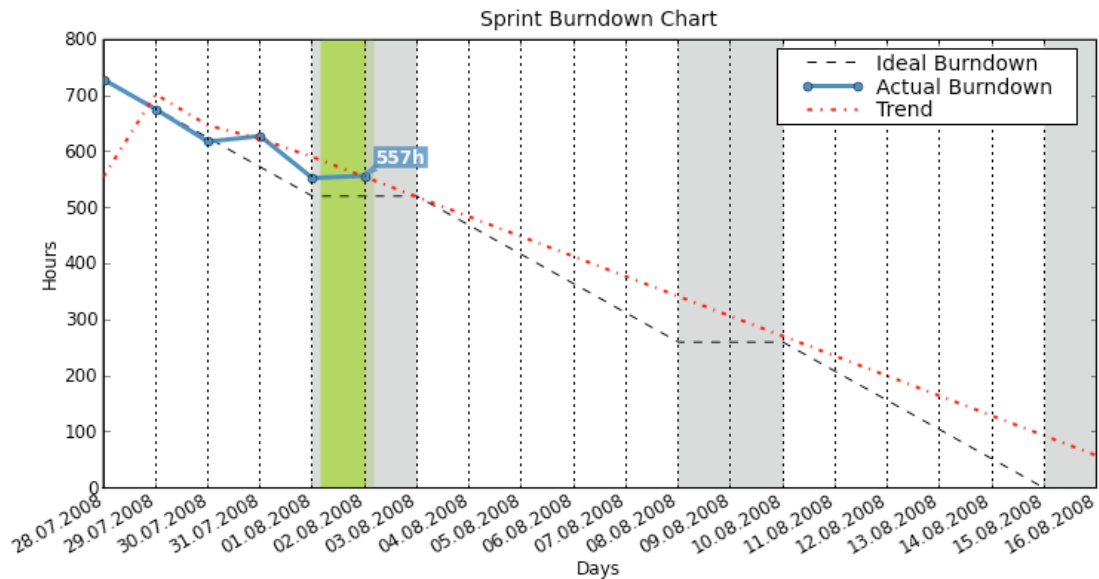


Figure 3.4: Burndown Chart. Chart showing the relationship between worked hours and planned hours in relation to the cost in hours.

3.2.1.4 Scrum over distance

The described method demands interaction on regular basis. This can be more difficult if the project is spread out over the globe. Kniberg (2007) rise issues of what communication bandwidth is needed in a spread out project. He lists the abilities that need to be covered and tools that can be used to cover them. Among the important tools to increase the bandwidth are: Web cameras, Headsets, "Remote enabled" conference rooms and windows and exchange programs.

He suggests two different approaches when organizing distributed teams: Either to split the teams according to their geographical setting or to make cross-location teams. For small teams he discuss that the cross-location teams is to prefer.

3.3 Previous studies of offshore software development and agile methods

There are several case studies done on how agile software development works in the practice. This section is built on four studies on software development in distributed and offshore environments and focusing on the

communication support within agile methods and the lessons learnt.

3.3.1 Background

Kussmaul, Jack and Sponsler (2004) have studied offshore outsourced software development. The study is committed on a distributed team with 2–3 developers onshore, in the US, and 5–10 developers offshore, in India. Another study was conducted on a company with part of their software development offshore in Bangalore. The development is further distributed between the US and Europe and the company has been using agile methods for the passed four years (Fowler, 2006).

Kniberg (2007) has noted down his lessons learnt when his company was one of the first big software developers to use the agile methods Scrum and XP. One part of their development is distributed.

Ambler (2002) has worked with agile software development in a big global company. He has recorded the results from a workshop on the subject of Dispersed Agile Software Development.

Jochen Krebs has played different roles in software engineering throughout his career, both on and offshore. In his three part article series he describe several different scenarios from outsourced offshore software development (Krebs, 2006).

3.3.2 Distributed offshore development

In order to make distributed development work you have to create relationships. The communication tools can help with this, but visits are needed. One way is to decide always to have at least one US employee offshore in Bangalore at all times, strictly for communication purposes (Kussmaul et al, 2004). Both formal and informal communication is important. To establish the informal communication between a distributed team, events such as kick-offs and releases help (Fowler, 2006). For example they start each Scrum meeting with some informal communication about

local news to help the team building. Kniberg (2007) describes tight collaboration and co-location of the Scrum team as the most important. Hence if there is a distributed environment the communication bandwidth has to be maximized. The following are listed as the most important to support:

- Pair programming
- Face-to-face daily Scrum meeting
- Meet physically and socialize
- Spontaneous meetings
- Common view of the project details

Fowler (2006) suggests that short iterations are more suitable, one-week is the perfect sprint length in distributed development. Ambler (2003) sees three major threats against using agile methods in an outsourced environment:

1. Time zone differences, not leaving enough room for synchronous communication.
2. Agile development is hard to organize over distance without losing the core of the methodology.
3. Cultural barriers make agile methods harder to implement offshore.

One lesson is not to outsource too small projects, this is not cost effective it is better to keep the development in the country. The suggestion is to start developing with local teams and as it grows the project can be placed offshore. All research, architecture work and requirements analysis should be kept close to the client (Kussmaul et al, 2004). To make distributed offshore development work flawless communication is central.

3.3.3 Communication

In distributed offshore development communication can be divided into two parts: Communicating the big picture in the project and everyday team communication (Fowler, 2006). Depending on if it is the teams or the team members that are separated: Different part of the communication is

important (Kniberg, 2007).

To support the overall communication and over bridge cultural differences Flower (2006) found ambassadors useful. The role of an ambassador is to travel between the offshore and onshore location and communicate both formal and informal information. This phenomenon is similar to the definition of grounding that Clark and Brennan (1991) describe (see 3.1.2).

Agile development also demands everyday communication and interaction. When the development is distributed the demands on communication are greater (Krebs, 2006). By implementing the following to improve the communication bandwidth (Kniberg, 2007):

- A web camera and a headset for each developer.
- Remote ready conference rooms (desktop sharing, conference microphones, web cameras)
- Streaming the remote location (video).
- Exchange programs between the locations.

Kussmaul (2004) implemented a shared mailing list for each team to which the members had to send a status mail everyday; this was done to spread awareness in the project. A daily IM meeting was also held. Synchronous and asynchronous communication is important for different purposes. Every project has to have communication “laws” and they should be implemented and forced by tools (Krebs, 2006). One should avoid one-on-one communication and rather spread the information by using multi-cast tools. One small thing like having a fresh IM status contributes with awareness to distributed teams (Fowler, 2006).

The minimum communication channels are to use a wiki page, IM and telephone. All channels are used for different purposes. When it takes more than to exchange a few chat lines to solve a problem there should be no hesitation to pick up the phone (Fowler, 2006).

Generally in software development a big part of the project work is communicated through documents. Key documents, detailed data models and data dictionaries are needed (Kusssmaul et al, 2004). The onshore and offshore teams need to have an agreement of the standards and expectations (Krebs, 2006). To ensure this common agreement it is important to make deliveries of ready functionality frequently (Kusssmaul et al, 2004). The best way to communicate design and interfaces is to deliver real graphical user interfaces so no misunderstandings will occur while interpreting the mock-up solutions (Kusssmaul et al, 2004).

4 Method

During six-month at Indpro's Bangalore office, different methods were used to form an understanding of the communication in the specific environment. The aim was to understand what support the method Scrum provides in the projects and to suggest a method to enhance the communication bandwidth within Scrum, using computer mediated communication tools.

The different methods were used iteratively and in different stages of the study. In the beginning the study had a wider approach and most time was spent to understand the whole environment through interviews and observations. Further on the methods addressed more specific issues concerning communication within specific projects. This part has been done through participatory studies, evaluations and structured observations. All methods are collecting qualitative data because of the environment observed is very particular and divided up into several small projects with different conditions. Figure 4.1 explains how the different methods were used over the time spread of the study.

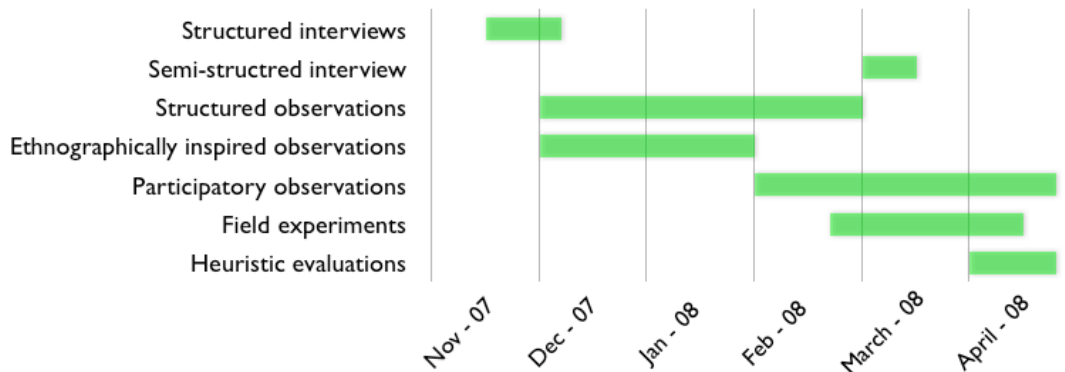


Figure 4.1: *Gantt schedule.* Schedule over what periods different methods were executed.

Below are more detailed descriptions of the methods used in the study: structured interviews, a semi-structured in-depth interview, structured observations, ethnographically inspired observations, participatory observations, field experiment and heuristic evaluations.

4.1 Interviews

Interviews can be done in many different ways. In this study I have used structured, semi-structured in-depth and ethnographically inspired interviews. Each was selected to suit the aim of the specific part of the study.

4.1.1 Structured interviews

In the beginning of the study I conducted interviews with all the software engineers presently working in the Bangalore office. The aim was to get to know the software engineers, their active projects and their knowledge and opinions about Scrum, but also to build trust and a relationship ground. The interviews were structured and all questions were written in advance and were asked face-to-face with one engineer at the time. The questions were divided up into two parts: One personal and one project specific part. The interview questions can be reviewed in appendix A and appendix B.

To collect the data notes were taken directly on the computer straight in the interview form (see appendix A). The subject of the interview was able to see the screen where the notes were taken. The reason for this was to smoothen the language barrier and simplify understanding. Although the interviews were strictly structured I let conversations drift somewhat, but only the answers to the questions were noted down in the form. In total 21 interviews were conducted and each lasted about 45 minutes.

4.1.2 Semi-structured interview

A semi-structured interview was held with an Indpro client, while he was working at the Bangalore office during six weeks. This was a semi-structured interview that lasted for about two hours. Semi-structured signifies when the interview questions are written in advance, but are of an open character and encouraged more free answers and the possibility to go deeper into one or a few questions (Bell, 2007). The interview questions can be found in appendix C. The aim was to obtain qualitative knowledge about the clients' project and the communication flow. Also to see the communication from the perspective of one of Indpro's clients was important.

The background information about the project was already retrieved by interviews with the software engineers (see 4.1.1) and through structured observations (see 4.2.1). Therefore this interview focused on more specific questions: It focused on the communication flow in the past and how it had been developing over the years and how the clients' presence in Bangalore had affected the communication.

Notes were taken during the interview directly on the computer as bullet points under a relevant question. After the session I went through the notes for another three hours to write out full sentences and clarify conceptions so that they wouldn't be forgotten.

4.2 Observations

Throughout the six-month long stay in Bangalore many observations have been conducted on regular basis. Two different types of observations were used: Structured and ethnographically inspired (Bell, 2007). The structured were used when the aim of the observation was known and more specific. The ethnographically inspired method was used when the objective of the observation was less specific.

4.2.1 Structured observations

The structured observations were used in the first half of the thesis work (see figure 4.1) when observing meetings between clients and offshore personnel and engineers. The observations consisted of listening on audio conferences. Some review meetings involved watching a streamed desktop during the conference. The observations were structured in such way that the focus was on the communication flow rather than the content of the meetings. One problem with using this method is the interpretation problem (Hansson, 2002). All observations are based on how the observer interprets what is said and done. To minimize this problem and get a higher reliability all results are always based on situations that occurred more than once in an observation situation.

To collect data, notes were taken throughout the meeting. If the meeting was audio only then the notes were taken directly on the computer. If the meeting contained visuals as well, demanding attention on the screen, then notes were taken down by pen and paper. All handwritten notes were typed into the computer directly after the meeting and all observation notes were sorted by date of the meeting. Later these notes were also coded by character and only re-occurring communication problems were saved.

4.2.2 Ethnographically inspired observations

Ethnographically inspired observations can be described as studies of an environment with no precise focus on the study. It is a spontaneous study or one with an unknown objective (Bell, 2007). The first period (see figure 4.1) in Bangalore was spent observing and learning the process and way of work without any active involvement. Surely there is a certain influence when adding an additional person to an office environment, called the influence problem. This should be taken into account when presenting and interpreting the result.

During this period notes were taken down on a computer, organized by date. The observation period was long and the notes had to be sorted out and coded gradually. Afterwards, only notes concerning communication issues remained. Although studying the entire process and the office was important in order to see the complete picture and an understanding of the environment.

4.3 Participatory study

During the second half (see figure 4.1) of the study the observations shifted into participatory observations, to be actively involved in projects as a team leader⁴. The team leader is in the center of communication of the projects at Indpro. Therefore this position was good while observing the communication flow. The position also gives opportunity to implement improvement ideas and evaluate how well they work. The aim of this part was to get the complete picture of how Scrum worked as a communication framework in this environment. Another aim

⁴ This role will be explained in 5.2.3

was to improve the communication through ad-hoc implementation of results and ideas from the theory.

4.3.1 Participatory observations

The observation part was done by careful note taking of what was done in the specific project. Due to taking on the role of as the team leader more focus were on the content of the meetings during this period. However, combined with the structured observations (see 4.2.1) this method should form a more complete image of the communication, which was the aim of the observations.

The collection of data was done by note taking and sorted by date and project. At the end of the day the notes were divided into two categories: The first with focus on the content of the meeting (i.e. meeting notes) and the second focusing on communication issues. Another data collection method was to capture the computer screen. This was done when there was no time to take notes or when an image was more suitable than notes. The screen dumps were filed sorted by project and saved with the current date as name. Furthermore I have saved other relevant documents, such as specifications and bug lists, in the project folders. The collected material also contains all emails received during this period, coded by date. I have gone through all data in chronological order when working with the results.

4.3.2 Field experiments

The aim of the implementation was to try different ideas that might improve the communication within the projects. The implemented ideas were based on the theory as well as on the results from previously done studies, described earlier in this chapter. After implementing the ideas the outcome was carefully kept track of by taking notes on how the new features worked in the process. All notes from this method were saved together with the notes from participatory observation (see 4.3.1), sorted in the same way.

Below is a list of implemented guiding principles for information awareness tested in a project as field experiments:

- *Shared mailing list.* To have all actors using an emailing list, so that all emails sent within the project goes out to all people involved in the project.
- *Meetings with all actors.* All actors in the project should be involved in all cross-location meetings.
- *All documentation on shared workspace.* To centralize the project information all documentation should be uploaded on the shared workspace.

4.4 Heuristic evaluations

At the end of the study (see figure 4.1) I made heuristic evaluations of the computer mediated communication applications used at Indpro. The aim was to get the pros and cons with the different tools and to give guidelines for what tools to use for communication in the future. The evaluations were done using the evaluation template “Ten rules of thumb for better usability” (Nielsen, 1994) combined with the approach of how the applications were used at Indpro. The rules of thumb are listed in Appendix D.

After the five months at the office I had a good idea, and practical experience, of how and to what extent the tools were used. I went through different fictive meeting procedures, documentation or communication scenarios, depending on what the application was used for. I noted down my thoughts about the different constraints listed in the ten rules of thumb.

5 Results

In the following chapter I will present the results from the methods executed as described in the previous chapter. The chapter is sorted after the characteristics of the results, not by method. During what period of the study a certain method was used is documented in figure 4.1. I will present results about the Bangalore office, Scrum implementation, Audio meetings, Collaboration, CMC and future collaboration.

5.1 Bangalore office

This section is based on the initial interviews and early structured observations to present background information on the office in Bangalore and the people that works there. The last section presents a brief description of the projects.

5.1.1 Software engineers

There were 21 software engineers working at Indpro's Bangalore office, all were born and raised in India. The mean age was 29 years and 90 per cent were male. All engineers had a degree from a university in India, eleven had bachelors and ten had masters. The average work experience as software engineers after university was four years. Ten months was the average employment length at Indpro.

5.1.2 Team Leaders

Two Swedish team leaders were working in the office. Both were male, 27 respective 29 years old. Both had master level education from a Swedish technical university.

5.1.3 Projects

Out of the eleven projects eight were using a product backlog and six were working in sprints. All projects were sending a weekly report to the product owner and all except for one team were sending a daily report to the team leader. One project had cross-location meetings involving both the team and the product owner. Four teams had regular planning meetings, with or

without the client. On average each project had 1.9 software engineering resources at the Bangalore office.

Table 5.1: *Projects A through K.* Table showing eleven Indpro project status in November 2007, demonstrating the projects size and what meetings and reports are used in the projects. (Abbreviation translation: BL= Backlog, SE = Software Engineers, PO = Product Owner)

Project name	BL use	Sprint	No. of SE resources	Daily report	Weekly report	Meeting with team and PO	Planning meeting
A	yes	yes	2	yes	yes	no	no
B	yes	yes	2	yes	yes	no	no
C	yes	yes	4	no	yes	no	yes
D	yes	yes	3	yes	yes	yes	yes
E	yes	no	1	yes	yes	no	no
F	no	no	1	yes	yes	no	no
G	no	no	2	yes	yes	no	no
H	yes	no	1	yes	yes	no	no
I	yes	yes	2	yes	yes	no	yes
J	yes	yes	2	yes	yes	no	yes
K	no	no	1	yes	yes	no	no

5.2 Scrum implementation

The interviews with the software engineers gave only a brief description of the projects. Combined with the ethnographically inspired observations on five of the projects (A – E in table 5.1) a more detailed description of how Scrum was used in the Bangalore office will be described in the following sections.

5.2.1 Documentation

In addition to the regular Scrum documentation and meetings the following documents has been implemented:

- *Daily Scrum report.* All teams sent a report to the team leader every day answering the questions:

- What have I done since the last Daily Scrum meeting?
- What will I do before the next Daily Scrum meeting?
- What prevents me from performing my work as efficiently as possible?
- *Weekly report.* On Fridays all teams sent a report on the weekly progress to the team leader who forwarded them to the product owner. The weekly report consisted of the following information:
 - Weekly report, a report on what has been done during the week.
 - Coming week overview
 - Weekly issue report
 - Weekly risk report
 - Miscellaneous
- *Sprint review report.* The team leader sent this report to the product owner after that a sprint was closed. The report consisted of the following information:
 - Sprint summary. A sprint report and issues raised during the sprint.
 - Sprint deliverables. Information about the tasks in the current sprint.
 - Sprint burndown chart.
- *MidSprint report.* The team leader sent out a status report to the product owner. The report had the same content as the sprint review report, with the aim of notifying the product owner of the sprint progress after half the sprint has passed. This was only done in some projects if the sprint length was two weeks or longer.

5.2.2 Meetings

The following meetings were added or redefined:

- *MidSprint meeting.* A meeting held between team leader and product owner. The agenda was to discuss the progress in the sprint so far. The meeting is the base of the MidSprint report (see 5.2.2)
- *Sprint review meetings.* This meeting had been developed as a

combination of sprint demonstration and sprint retrospective. The meeting agenda varied in different projects, but were generally focusing on what tasks had been done and not, and the reason for this. The meeting was the base for the sprint review report.

- *Coffee break Meeting.* A daily meeting for geographical separated team members to share informal information. The only rule in this meeting was not to talk about work and it was only used in project D.

5.2.3 Roles

The following roles are added or redefined:

- *Team leader.* A Swedish person in Bangalore who was in charge of the Scrum process. The team leader was also an ambassador and should over bridge the cultural differences between Indians and Swedes.
- *Project leader.* A Swedish person on the product owner's side who worked in close collaboration with the team and the product owner. The project leader could have been an independent consult or working at the product owners office.

The team leader and the project leader shared the responsibility as Scrum master to represent the team to the product owner and vice versa. In the projects without project leaders, the team leader had the full responsibility to do this and the role is more like the defined Scrum master (see 3.2.1.1), but with an addition of ambassador responsibilities.

5.2.4 Scrum in practice

Table 5.2 is a chart of the communication dimensions of the projects that I have looked more closely on in the structured and ethnographically inspired observations and structured interview. The projects signifies the same as the projects A – E in table 5.1. The five projects I chose to look more closely on were different in size, character and Scrum usage. Four out of five projects used sprint backlogs (i.e. worked in sprint iterations) and two had a product owner involved in the sprint planning. In three projects the client had an assigned project leader to run the project.

Table 5.2: Project A through E. Table over the status in five Indpro projects in January 2008 demonstrating the clients involvement and what CMC are used. (Abbreviation translation: PO = Product Owner, TL= Team Leader, BL= Backlog, W= Weekly Report)

Project name	PO involvement	Resources	Scrum meetings	Documentation	CMC use (PO/PL and team/TL)
A	Daily	2 in team 1 TL 1 PL 1 PO	Daily scrum, Sprint planning, Sprint review	Project BL, Sprint BL, Burndown charts, Daily Scrum, W	Skype (audio) E-mail Sharepoint.
B	Weekly	2 in team 1 TL 1 PL 1 PO		Product BL, Sprint BL, Burndown charts, Daily Scrum	Skype (IM/audio) Email Sharepoint
C	Monthly	4 in team 1 TL 1 PO	Daily Scrum	Product BL, Sprint BL, Burndown charts, W	Email
D	Daily	3 in team 1 TL 1 PO/PL	Daily Scrum, Sprint planning, Sprint review, Mid-sprint	Product BL, Sprint BL, Burndown charts, Daily Scrum, W	Skype (IM/audio) Sharepoint Email
E	Daily	1 in team 1 TL 1 PO		Product BL, Daily Scrum, W	Skype (IM/audio) Email Sharepoint.

In three projects the product owner had contact with the team through the SharePoint environment. All product owners used email to communicate with the team leader. One product owner had direct contact with team using Skype, the rest chose to contact the team leader or project leader instead. In two additional projects the project leader had direct contact with the team using Skype.

Project D had during my first months in Bangalore grown from three engineers to nine. The same client was now the product owner for three offshore teams, each with three software engineers. These projects will further be referred to as D1, D2 and D3. The interview subject in the semi-structured interview (see 4.1.2) is the project leader, Jonas Eriksson, for the original team described (now: D1), but was in charge of all of the client's projects while working in Bangalore. This would also be his main responsibility when returning to Sweden. In order to do this effectively he was going to start working Indian hours when he returned. Eriksson felt as if it was important to have a Swede working in close collaboration with the Indian teams. This was his second stay at the Indpro office; the first visit was six days long and this time he was staying for 45 days. In the future he plans to travel to India as often as possible.

The teams had different conditions for their collaboration with the clients. D1 was not distributed and the Indpro team was the only team working on the project. The second team (D2) was working as cross-location team members, in close collaboration with the team members in Sweden. The third team (D3) was working distributed together with other teams in Sweden, a method known as geographically separated teams.

Eriksson thought Scrum had improved the collaboration much. Although he thought the implementation of the method had been difficult. The agile concept had been hard for the Indian software engineer to grasp. During his first visit he was able to root the model in the original team, and after that the teams has taught each other. Over time, the client has learned to use a sharper version of Scrum and give more specific directions and write the specifications more carefully.

Other issues observed when using Scrum in the Bangalore office:

- Items were pushed in to sprint as they come up, i.e. the sprint was never completely locked.

- No time was set aside for estimation of the product backlog when requirements changed.
- Sprint length was unclear for the team, why they only populated it with the tasks that they were currently working on.
- The sprint end date was not set in the beginning of the sprint, i.e. the end date was set ad-hoc when sprint tasks were done.
- No review or retrospective meeting of the sprint was held; the only review was the sprint review document.
- The teams didn't have a daily scrum meeting; they putted together a daily scrum report by sending the information to each other over Skype IM.
- Only project D had proper demonstration meetings where the teams presented their work.

5.3 Synchronous Communication

This section is based on the observations executed in the second half of the study, structured observations. The focus of this part of the was the synchronous communication between the two locations, both in audio meetings and instant messaging.

5.3.1 Audio Meetings

The structured observations was committed on the same five projects A – E (see figure 5.2). This study contributed with more detailed information on how meetings were deployed and how Skype's audio link was used at the Bangalore office. The observed problems were divided up into technical issues and user related in the following sections.

5.3.1.1 Technical issues

Several times when starting a meeting participants were not able to speak or hear the conference. Problems such as a broken headset, a not connected headset or the settings in the Skype application were wrong occurred frequently. Sometimes the cause of the problem was unknown, but could usually be solved by re-call all conference participants. This happened for both participants in Sweden and in India. In project D all software engineers had their own headset,

always connected to their computer. In one clients office, in Sweden, they had a separate conference room with a conference microphone.

Digital disturbance often occurred during Skype conferences, problems such as robotic sounding voices or that sentences and words disappeared during a meeting. Another observed incident were that surround sound from the office disturbed the conference. The sounds were coughing, beeps or loud voices. When conference participants were in the same room their voices gave echoes through double recording in the different microphones. Often the mute function in Skype was used to prevent this problem. Then all participants, except the one talking, muted their microphone line-in. This made the virtual conference room very quiet. Several times I observed that the person talking was uncomfortable, due to the lack of feedback in this quiet room. Using the mute function also lead to that conference participants were talking while their microphones were on mute and had to repeat what they just said.

Other technical issues with Skype observed in the study:

- The conference participants started talking at the same time, due to latency.
- One of the participants had a computer that crashed and missed part of the meeting.
- An internal application failure occurred and it was not possible to connect with one or more participants.

5.3.1.2 User related issues

Often conference participants missed or misunderstood something that was said during the conference. This applied on both the clients and the developers. The insecurity of what had been said in the meeting lead questions to the team leader after the meeting or that the team leader had to clarify and explain during the meeting. In the interview with Eriksson, he described his relationship with the offshore developers. Before he had visited India the first time he had a hard time to understand the developers and the team leader often had to translate. After his first brief visit he suddenly started understanding everything.

Along with him the other co-workers in Sweden also understood the Indian coworkers much better, and vice versa. All meetings suddenly got a much better flow. Eriksson said: "It felt like the developers no longer thought of me as a scary foreign person, but somebody just like them"⁵. During the first visit he shared informal information from the Swedish coworkers, among others a video from a company party. His experience told him that informal communication is the key for a successful collaboration. At this point the software engineers were no longer afraid to express themselves to the Swedish teams or ask for help when they need it, this had previously been a problem. He thought it was important with ambassadors and was looking forward to being able to send Indian engineers to Sweden, and vice versa.

Non-verbal communication while having muted microphones was another behavior observed in the Bangalore office. The team members often discussed with each other, outside the conference, what was said in the meeting. They also used other physical communication, such as nods and winks, not visible in the audio conference. In the future Eriksson wished to see more streamed live video from the Bangalore office.

Other user related issues observed in Skype meetings:

- The meetings did not have an appointed moderator. Problems such as a silence after an asked question, it was unclear who, if anybody, is supposed to answer.
- No agenda set up for the meetings. Sometimes an agenda was created ad-hoc in the Skype IM with all participants. This agenda was never saved anywhere.
- During the conference IM was used vigorously. The participants were sending links, comments and clarifications to one-another. Especially when having bigger meeting and muted microphones.

⁵ Authors free translation from Swedish to English.

5.3.2 Instant messaging

Skype was also used for communication outside of meetings. In two of the projects the project leader used the IM function to get in contact with the team, and the product owner used it to get in contact with the team leader. Generally an IM conversation was used to initiate contact and then if the issue was not solved in a couple of lines then an audio conference was started or planned. IM was also used between the team and the team leader, although they were in the same physical space. Also between team members the IM function was used, and also between coworkers in the Bangalore office. When chatting both formal (such as asking for help from other developers/team leaders) and informal (such as planning a tea break with coworkers) information were exchanged.

The common way to share documents in the Bangalore office was to send them over Skype. Even between Sweden and India, Skype was often used to share documents more often than email or SharePoint.

In project D communication between developers in India and Sweden was held continuously throughout the day. For this the IM function in Skype is used. Eriksson did not think this was optimal; he wanted to be able to see the developers and the office while he was working in Sweden.

5.4 Asynchronous Communication

The participatory studies focused more on the asynchronous communication in the projects. The use of the collaborative workspace SharePoint was central, but also the use of email was studied. At this point of the study the involvement was narrowed down to the three projects (A – C) showed in table 5.3. These are the same projects A – C as described in the previous sections in the result chapter.

Table 5.3: Project A through C. Table showing the communication details of three projects in March 2008. It demonstrates what CMC tools are used for in the Scrum process. (Abbreviation translation: PO = Product Owner, TL= Team Leader, BL= Backlog, PL = Project Leader)

Project name	Sharepoint use	Email use	Skype audio use	Skype IM use
A	Product BL, Sprint BL, Burndown chart, Team discussion, Bug list, Shared documents	Various kinds of conversations between PO and TL over email.	Meetings between PO/PL and TL.	
B	Product BL, Sprint BL, Burndown chart, Team discussion, Shared documents, Calendar	PO sends lists of to-do items and bugs to TL.	Meetings between PL and TL	Sessions for clarification between PL and team/TL.
C	Product BL, Sprint BL, Burndown chart	Between PO and TL.		

5.4.1 Collaborative Workspace

SharePoint is a collaborative workspace on the web⁶. Indpro used one individualized workspace set up for each project. Only the people involved in the project had access, and different actors had different rights in the interface.

As a cooperative workspace SharePoint laced in feedback and awareness of what was going on in the workspace. It was possible to see what was new in a list through a “new” image, but if something is added in a folder the change was invisible. Further the workspace gave no awareness of others logged in users. Two people could edit the same post simultaneously.

At Indpro the SharePoint server was located in Sweden and therefore the response times were high, i.e. it took a significant time to upload and save content in the workspace. This led to another observed issue; the software

⁶ More information about Microsoft SharePoint can be found on <http://www.microsoft.com/Sharepoint/>

engineers hesitated to add content to the workspace and rather shared the files over email or Skype.

5.4.1.1 Product and Sprint Backlog

All observed projects used SharePoint to hold their product backlog, sprint backlog and a burndown chart. The backlogs were represented as lists in the shared workspace, what content they had vary from different projects. Common in product backlogs were:

- ID
- Title
- Work order
- Status (Select box with the following options: Under construction/Constructed/Requirement changed/Ready/In Sprint/Pushed back/Completed/Removed/Problem/On hold)
- Planned sprint
- Time estimation (Counted in days or hours)

An observed attitude was an insecurity of where to add changes, they had been randomly added as product backlog items, sprint tasks, and in shared documents (see 5.4.1.4). When new items were added to the product backlog in the middle of the project they got the status constructed. When the team thought that the provided information was inferior they posted their queries in the team discussion and changed the status to "Under construction". When everybody had agreed on the terms of the item the status was changed to "Ready" along with time estimation.

Another issue was that one team used a trial and error, iterative, approach. They marked the item as ready, although they felt as if the information was not sufficient, and started develop a solution. They knew their solution wasn't complete, but did announce the item as complete anyway.

The common sprint backlog columns were:

- ID
- Title

- Assigned to
- Status (Select box with the following options: Not started/Work in progress/On hold/Problem)
- PB Item (Selection of Product backlog items)
- Hours remaining
- Comment

In one project the column “actual hours” was added and used in the list. When the team had questions about a task they post this as a comment in the sprint backlog. The comment often resulted in an asynchronous dialogue between the team, project leader and product owner. In the comment field any actor could erase the content and there were no version control in the field. This resulted in that data had been deleted from this column at times.

The study has shown that sprint backlogs were not always populated with the proper amount of hours in the beginning of the sprint. Sometimes it was because the team didn't know what should be included and only added the things they were working on now or the entire product backlog item was put in the sprint even if it is too time consuming for the particular sprint. Extra hours were added to a sprint, to cover if something went wrong.

One problem was that planning a meeting was held between product owner, project leader and team leader in Swedish. Afterwards the team leader would forward the decisions to the team by locking product backlog items in the upcoming sprint. Often the team didn't agree decisions (for example another item has to be done before a locked one), which led to that the sprint had to be re-planned. The team leader then had to notify the product owner and the project leader about this and get their feedback before locking the sprint.

5.4.1.2 Burndown chart

All projects also had product and sprint burndown charts in the workspace. Throughout my observations I did never see or hear about anybody in the projects using the charts. The only exception was when the team leader was putting together the sprint review reports (see 5.2.1) where the charts were included. Whether the product owner took notice of the charts was not clear,

although it has never been up in discussion in any of the meetings I have attended. The team leader updated the burndown chart daily. If the team didn't update the sprint backlog daily, this left a misleading chart. If the sprint wasn't populated and locked properly in the beginning of the sprint, the outcome was false.

5.4.1.3 Team Discussion

Two of the projects were using a team discussion in the workspace. This was an asynchronous thread forum. It enabled all actors in the project to post text and links, the forum was threaded and other actors could reply to the posted items.

The team discussion was used to refer to other parts in the workplace environment by posting links to documents. Also observed was that parts of IM conversations between different actors in the projects were pasted in the forum. A problem was that when an answer was posted in a thread the person that posted the question didn't get notified, which slowed down the communication flow.

5.4.1.4 Bug list

A bug list was a similar listing as the backlogs. It was used by one of the observed projects and contained the following columns:

- ID
- Title
- Assigned to
- Status (Select box with the following options: Active/Resolved/On hold/Clarification needed/Closed)
- Priority (Select box with the following options: High/Normal/Low)
- Due date
- Bug type
- Estimated hours
- Modified

Far from all bugs were added to the list. At times the product owner sent bugs in documents over email to the team leader and the team. The team leader

sometimes putted these bugs up in the Shared document (see 5.4.1.5) or forwarded them to the team. At times bug list items were created and referred to these documents. It seemed hard to distinguish between new product backlog items, changed requirements and bugs. This created a confusion of where to add the issues.

When new items were added to the bug list the team was not notified about it by the workspace. It was not possible to add bugs to the sprint backlog (the list demanded a product backlog item), to solve this bug list items were pushed in sprints through adding extra time, as described in section 5.4.1.1.

5.4.1.5 Shared documents

Shared documents was a file sharing area. All actors in the project could upload and download documents and files here. The text documents could be edited in the browser if Internet Explorer was used, otherwise a version control could be enabled. Documents and other files could be structured by the use of folders.

The shared document space lacked feedback. If something was added inside a folder this was not displayed anywhere and no alert went out. In a project with an unorganized document structure I observed team members having to spend much time looking for a specific document.

5.4.2 Email

Email was used for asynchronous communication between product owner and team leader in all projects. The content of the emails varied, everything from task specific questions to documents with meeting agendas and chat-like conversations. Emails that were sent to the team leader could be a question directed to a specific team member. Other times questions was sent to one team member, when it actually concerned several more actors in the project.

5.5 Future collaboration

In the study it became clear that one problem was the centralization of the project information. Different actors in the projects had different information and it was unclear where and what information should be shared. In the

participatory field experiment some information centralization help was introduced in project A, the goal was that everybody involved in the project should have the same information. To do so a few ideas were tried in the project. Presented below is the result from the implementation of a shared mailing list, having meetings with all actors and adding all documentation to SharePoint.

5.5.1 Shared mailing list

The first thing to be implemented was a shared mailing list. When using this email address all actors in the got the sent mail. All email errands⁷ concerning the project should now be sent to this list, even if the receiver is one specific person. All replies should also be sent to the list. All emails should be written in English.

The shared email list increased information sharing between the two locations. The team got more information about the project, and felt more involved and motivated. It also increased the direct communication (dialogue-like) between team and product owner. It opened a new channel, which is somewhat more synchronous than SharePoint because the actors check their mail more often than they use SharePoint.

Another result was that the shared email list got overused, one example is that most documents were sent over email instead of added to SharePoint. As another aim was to gather all information on SharePoint (see 5.4.1) and use the email list only for issues that needed quicker response, such as meeting planning and critical errors.

A new use for the email list was to complement the lack of the awareness in SharePoint. When new things were added or posted in SharePoint a message that this had been done was sent to the list.

⁷ A email errand is an errand that was previously always sent by email to one or several actors in the project.

5.5.2 Meetings with all actors

The aim of the second experiment was to include all actors in all meetings to shorten the communication steps, described as a problem in section 5.3.1.2. The different meeting types held in this project were review and planning meetings. They were usually held the same day with a shorter break in between.

The planning time for the meetings increased in order to fit the meetings into all actors' schedules. The connecting time also increased, as more actors were to attend the meetings this resulted in more technical problems occurred (see 5.3.1.1 for details). One time a meeting had to be re-scheduled due to technical difficulties that took more than 40 minutes to fix.

The meeting duration decreased, but additional meetings were still held between team leader, project leader and product owner. The reason given for having additional meetings was that the issues to be handled were easier to discuss in Swedish and that the team input was not necessary. Although these meetings often resulted in new questions to the team, either posted in SharePoint or cleared out with the team by the team leader.

The team did not feel comfortable in the meetings. Many misunderstandings and language barriers similar to the once explained in section 5.3.1.2 were observed. The team rather asked the team leader questions after the meeting if they felt that something was unclear.

Throughout the study less and less meetings were held with the team. The request not to involve the software engineers on the meetings was a request from the product owner, who thought the meetings were too complicated and ineffective when they were held in English with all actors.

5.5.3 All documentation on shared workspace

Another experiment with the aim of centralizing the information is to force information to the shared workspace. One new arrangement was to add the

daily scrum report (see 5.2.1) as a list in the workspace. I created a simple list with the following columns:

- Current Sprint
- Since last Scrum meeting
- Until next Scrum meeting
- Issues

This way the product owner and the project leader also could take part in the daily scrum report and not only the team leader. Also it transferred traffic from email to the shared workspace. The list resulted in more awareness of the process for the product owner. The team felt that it was an improvement to do more things at the same place.

In order to centralize information in SharePoint I started to submit a meeting protocol to SharePoint after each meeting. But it had little effect since neither the project leader, product owner or team downloaded the document. Later the meeting protocols were sent to the shared email list, which seemed to share the information better, i.e. more actors take part of it. Another advantage is that it is possible to comment on the meeting protocol when it is sent over email. A general improvement was to start using meeting protocols; this helped to spreading the information between the actors.

As previously discussed (see 5.4.2) most documents was sent over email instead of added to the workspace. When starting this experiment I added all documents that was sent by email to SharePoint and referred to them instead. But when adding a document to the workspace you also needed to send a notification email to the shared list it was hard to motivate this double work. Over time the shared documents were only used for really big documents all the rest was sent through email.

6 Discussion

The discussion presents my opinion of the results and highlight what I think are the most important lessons learnt while studying this environment. The discussion covers the results, method, theory and future work.

6.1 Results

The result chapter is an objective description of the communication channels in the Bangalore office. In the discussion of the results I will focus on the other two research questions:

- What communication support does the agile method Scrum provide for an outsourced–offshore environment?
- What collaboration technology is needed to enhance the communication and bandwidth?

6.1.1 Communication and Scrum

The results show a lack of in communication and understanding between the stakeholders in Sweden and the teams in India. There is a great restrain in the communication flow between the two locations. Little or no time has been put aside for the software engineers to get to know the Swedish clients. It is the team leaders role to be responsible for that the meetings between the different actors should be seamless and to clear out any misunderstandings. It is not only to get the Indian software engineers to understand the Swedish way and the vision of the product owner, but time should be set apart to give the Swedish clients a better understanding of the Indian culture and way of work.

The agile manifesto states the importance of individuals and collaboration, which is also highly valued when working with Scrum. In order to collaborate efficiently over a distance, common ground and awareness are important. To have all meetings with all actors was an unsuccessful experiment in this study. The failing result might be caused by the shortage of time that was put aside for informal information but also because the methodology was introduced in the middle of the project. To emphasize informal information exchange is

important, especially in the beginning of a project, so that the client can share his whole vision with the team. A unified vision will help to create a common goal to work towards. In Scrum no informal kick-off meeting is defined but it is central to add to an outsourced offshore environment.

To support informal information exchange between the different actors throughout the project, one meeting need to highlighted more at the Indpro office: the Coffee Break Meeting. This meeting was only used by one project and I see a need to apply this meeting to the other projects as well. However, the meeting doesn't necessarily have to be a stand-alone meeting, but daily time should be put aside to discuss non-project specific issues. It can be short, but should be as prioritized as all other meetings. Another way to support informal communication is to use ambassadors; this has clearly improved the collaboration in project D. One problem is that most Indpro projects are small, with an average of less than two offshore resources, and it might be seen as a big expense for the client to travel overseas. The use of Swedish team leaders are working well as ambassadors, but might also stand in the way of client's direct collaboration with the team in some situations.

In the experiment the software engineers also took too little space in the meetings, even so little that the other actors thought there was no point in including them in the future. One solution could be to have refined demonstration meetings, where the developers go through and demonstrate what they have finished during the sprint, while product owner and project leader can ask questions. This meeting would enhance the groupware awareness, which can compensate for the lack of common ground. Groupware awareness is strictly focused on the task and not on the people executing it. In the demonstration meetings the software engineers would be forced to take a more active part of the meetings, and get a natural starring role. A common ground would consist in the application they are demonstrating and should therefore make the communication more flawless. Also to get direct feedback on the work would be a foundation for the actors in understanding what each

other think. Direct feedback, positive and negative, is also a good motivation for the team.

To get the most out of Scrum in this environment some of its agility has to be sacrificed. Sprints have to be more harshly locked, have a fixed sprint length and pre-decided meeting times. A stricter framework makes the collaboration easier, because all actors involved know what is expected of them. Since the method was introduced in mid-project it is difficult to get it right immediately, this demands constant follow-up and reviewing. This means that the way to work is new to both the team and the client and if the rules are too loose it is easy to fall back to the old way to work. This problem will be eliminated when projects start with Scrum from the launch. To start reflecting over the method on regular basis is important in order to develop a working solution individualized for each project. The Scrum retrospective meeting should be introduced at the Bangalore office in all projects. During this meeting the method should be evaluated and all the actors should be involved to improve the Scrum usage. It is a time to review what is working and not working. Also to discuss misunderstandings and the Sprint Burndown Chart. The meeting should focus on giving the developers time to reflect on the work and describe what they need, in order to make the collaboration better.

6.1.2 CMC Usage

Technical problems are also an obstacle in the collaboration. The technical problems during audio meetings are often caused by the infrastructure in Bangalore and in the office. Some of the local technical issues are caused by the shortage or bad quality of headphones in the office. These are things that can be improved by investment of new, better equipment. Another suggestion is to have a remote communication enabled conference room. To use conference microphone and speakers would decrease the start-up time of the conference and demand lower digital bandwidth, but can also lead to new problems that need to be studied. Technical solutions such as videoconference systems could increase the awareness, but the current infrastructure in the Bangalore office doesn't support such digital bandwidth demanding techniques. This is also a large investment and the gain in awareness might not cover the cost in money.

Another usage of video technique is to stream the whole office. By doing this some non-verbal communication would be somewhat translated to the actors based in Sweden. But more importantly it would provide an additional communication channel and build awareness through transmitting more subtle information from the office. This type of awareness would strengthen the collaboration, even though the video quality is too poor to stream facial expressions it would give the observer a better understanding on what was going on in the office. To open such window into the office could help create a better understanding of the workplace context for the clients, which is important. Another way to support awareness is through use of avatars, representations, of the actors in the projects. The goal is to be aware of what one another is doing at all times. To have a fresh status on Skype might facilitate this problem. It is important to remember that this goes both ways; if the software engineers should keep a fresh status then the client should also be obligated to do so.

Collaboration is not only important in meeting situations, but also in everyday work. One very important communication channel is the shared workspace. Using a shared workspace with good support for awareness can compensate for a missing common ground. The results show that the way SharePoint is used at Indpro gives poor support for both awareness and collaboration. Well working workspace awareness should give information about what is happening in the system, it should be focused on the executed tasks and not the people working in it. SharePoint lacks in both workspace awareness as well as awareness of the people using the system. The results indicate that there is a need for awareness from both sides of the development. The product owner want to be aware of what the engineers are working on and the engineers want to know what the product owner considers about different issues.

There should be a benefit and not an effort to add documents to the shared workspace, which is currently not the case. The users have to sacrifice valuable time looking for documents and adding documents to the workplace. One

benefit of adding documents to the workspace could be that they are dynamic and give the different users a possibility to comment and edit. This is a function supported in SharePoint, if using a certain browser, but not used at Indpro. In order to get the actors to use the workspace stricter communication rules should be set up and forced on its users. There is also a need for stricter roles in the workspace. It should be clear who is responsible for what in the workspace. Although both problems could be solved by a different method: assigning the problem of availability. The shared workspace should be as simple as possible and tailored for the work process. It should be simplifying the work for all actors in the projects and be the first option when collaborating and searching for information about the project.

Also missing in the workspace is a place to leave non-task specific messages to the other actors. An open message board would compensate in many ways for emails and IM discussions and result in a more centralized project communication. If the workspace encourages collaboration and bonding it will contribute to the teambuilding. Together with a list of recent activity in the workspace and awareness about the users in the system, the collaborative effort would decrease.

6.2 Method

The assignment given by Indpro was a general invite to study the communication within Scrum methodology in their offshore-outsourced environment. This demanded that I had to properly know the culture and how things work around the office. Therefore this was the focus in the beginning of the study. The time spent on the Bangalore office was nearly six months and was a combined study and employment, an internship. In a dynamic and developing environment such as the Indpro office plenty of things changed over this time. Observations and experiments were committed throughout at different stages of this development and the basic conditions are therefore unlike for the research methods, which is an obstacle when presenting the results. I found more value in doing it this way, rather than to execute my entire study on full time in the beginning of the internship. The longer I stayed the

more deeply entered knowledge I accumulated about the environment. The field-orientated study also enabled me to add and complement my study over time.

To be a part of the team as a team leader with active involvement did affect the results, if another person would have been there in my place the outcome could have been different. Another person with different competences and more experience in the company or in the role as a team leader would have another influence on the development. When I started the study I had no preconceived notions about the process and took in all information with fresh eyes and an open sense, therefore I could identify situations more openly than somebody that was more involved.

The decision to be a part of the team and not only observe it was made because I wanted access to as much information about the communication flow as possible. Much information is implicit and can only be understood if a full understanding of the environment is obtained. The choice of collecting data only by notes, images and documents can be questioned. This way much information can be lost this way, but the entire amount of notes etc. are great and to also analyze and transcript recordings would be too time consuming for this study. Since I was working in the field I had access to supplementary information at any time. I could go back and ask people questions about attitudes towards changes and process specific things, when things were unclear.

6.3 Theory

The study confirms many of the results described in the section with previously done studies (see 3.4). The previous studies mostly focuses on the technical issues that might occur in these types of environments. The ideas for the field experiment, such as the usage of a shared mailing list, did derive from the theory (but was executed differently). Also the background about CSCW has supported me to look further into the informal information exchange, through this I acknowledged the shortage of awareness between the actors. In my result

I stress the importance of the human aspects when mixing cultures and using mediated communication. It is important to get a clear understanding of the users, which is the base of understanding why a message get misinterpreted by it's receivers. The core is to look beyond the problems with for example to little digital bandwidth, something that will be solved over time, and see how relationships and a trust can be built up so that the understanding between the actors reaches a higher level.

6.4 Future work

In the future more effort should be put to create an informal relationship between the team in India and the client in Sweden. This should be especially stressed in the launch of new projects in order to create an initial understanding and common ground between the different actors. Forcing and planning meetings are critical in an early stage of the project, the meetings should be both formal and informal, and should result in a dual directed trust and understanding between the Swedes and Indians. This support could also come through using a sharper version of Scrum with stricter and continuous meeting times and habits. More evaluation is needed, especially in the launch of projects to form a working collaboration foundation. All procedures should be clear to all actors before starting the development.

Another improvement would be to invest in better technical equipment such as a remote enabled conference room and permanent and high quality headsets at each workstation. In order to improve the awareness and collaboration, this should have better support in the shared workspace. The main purpose for the workspace should be to support, not a place to report. Find ways to use the workspace so that it is more sufficient than other CMC's for all actors and force this way on the users. By setting up communication laws users might be troubled in the beginning, but if the laws are beneficial they will realize this and the collaboration will be superior.

7 Conclusion

I would like to divide my most important results up in three different categories: Scrum usage, awareness support and hardware improvements. Each of these categories addresses several issues that need to be improved in order to attain a better communication flow when using Scrum. The categories will be explained below along with a bullet list intended as a concrete basis on things to continue work with at Indpro AB.

7.1 Scrum Usage

The Scrum methodology provides a framework for collaboration in the outsourced offshore environment at Indpro. However, additional meetings need to be implemented so that the client and the teams can form a closer bond and collaborate better. In order to build up a close collaboration the informal information is essential. To build trust between the actors they need to share a vision and feel comfortable with each other. To implement informal meetings and emphasize on direct feedback is important to establish trust. My suggestion to be able to achieve this is by introducing Coffee Break meetings and reorganizing the Demonstration meeting so that the software engineers are moderating it.

I also suggest that Scrum has to be more closely regulated and more meetings and documents should be forced on the client and the development teams. The software engineers have to be given freedom within the framework so they can see the advantages in using Scrum, such as being able to work without interruption in a sprint. The communication rules have to be clear to all actors when starting the development.

7.2 Awareness Support

The communication flow in the Bangalore office could be improved by amending the shortage of awareness and the decentralized information within the projects. Both of these are connected to the shared workspace. In order to centralize the information the platform should be encouraging and using it should be beneficial for all actors in the project. The workspace should also, as

opposed to how it is used now, give its users feedback, status of the project and information on what everybody is working on. This way a workspace awareness can be created.

Other ways to support awareness is to have a fresh Skype status and or by using avatar representations of the people involved in the projects. Ambassadors should be used more. Both sending Indian software engineers to Sweden and the Swedish clients to the Bangalore office is going to create a better awareness and simplify the grounding in communication.

7.3 Hardware improvements

Minor improvements in the infrastructure could make great improvements for the meetings: such as investing in a conference microphone and speakers would make the meetings more of a social event and tighten up the collaboration. To provide the software engineers with more directed microphones, i.e. better headsets, would also ease the communication flow. Another way of making the client more aware through a hardware improvement is to open an additional communication channel by streaming the Bangalore office.

7.4 Future work

The following four bullet points are my recommendations for future communication work on Indpro:

- Facilitate to form informal relationship between client and team by implementing Coofee Break meetings on all projects and revision the Demonstration meeting
- Improve awareness and collaboration in the shared workspace by forcing communication rules that engage all actors
- Sharpen and restrict the Scrum method with rules and guidelines
- Invest in better headsets and a remote communication enabled conference room

References

The Agile Manifesto. (Electronic). Available: <<http://agilemanifesto.org/>> (2008-08-18)

Ambler S. (Electronic) "Bridging the distance" IN: *The Agile Edge, Sept. 2002*
Available: <<http://www.ddj.com/architect/184414899>> (2007-12-07)

Andersson P. J. (2006) "Moderna Indien" (Modern India) Alfabeta Bokförlag AB: Stockholm.

Bell J. (2006). "Introduktion till forskningsmetodik" (Introduction to Research Methodology) Studentlitteratur: Denmark

Clark H. H. & Brennan S. E. (1991) "Grounding in communication". In: *L.B. Resnick, R.M. Levine, & S.D. Teasley (eds.) Perspectives on socially shared cognition*. Reprinted in: *Baecker (ed.) (1993), Readings in Groupware and Computer-Supported Cooperative Work*.

Extreme programming (Electronic) Available:
<http://www.xprogramming.com/what_is_xp.htm> (2007-11-02)

Eriksson Fogh I. (2005). "Det indiska IT-fenomenet" (The Indian IT Phenomenon). Sveriges Ambassad : New Delhi

Fowler M. (2006) "Using an Agile Software Process with Offshore Development" (Electronic) Available:
<<http://www.martinfowler.com/articles/agileOffshore.html>> (2007-12-07)

Gutwin C. & Greenberg S. "A Descriptive Framework of Workspace Awareness for Real-Time Groupware" In: *Journal of Computer Supported Collaborative Work*. 2002:11. Pp. 411 - 446.

Hansson S. O. (2002). "Konsten att vara vetenskaplig" (The Art of being Scientific). KTH:Stockholm.

Indian Embassy. "Indian Facts" (Electronic) Available:
<http://www.indianembassy.se/index.php?option=com_content&id=54>
(2008-05-12).

Kniberg H. (Electronic) "Scrum and XP from the Trenches." Available:
<<http://www.infoq.com/minibooks/scrum-xp-from-the-trenches>> (2007-11-02).

Krebs, J. (2006) "Taking Off to the Smart Shore" (Electronic) Available: <
<http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=ART&ObjectId=11523&tth=DYN&tt=siteemail&iDyn=2>> (2007-12-07)

Kussmaul C., Jack R. & Sponsler B. "Outsourcing and Offshoring with Agility: A Case Study." In: *Extreme Programming and Agile Methods*. 2004. Pp. 180 – 188. Heidelberg : Springer Berlin.

Ministry for Foreign Affairs (Electronic) "Indien". Available:
<<http://www.regeringen.se/sb/d/2520/a/13973>> (2008-05-22).

Nielsen J. (1994). Heuristic evaluation. In: *Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods*, John Wiley & Sons: New York, NY.

Olson G. M. & Olson J. S. (2000) Distance Matters. In: *Human-Computer Interaction*. Pp. 139-179.

Overby S., Wikberg J., Heymonwska A. & Röhne J. (2008). "Global Sourcing 2008: Slaget om Europa." (Global Sourcing 2008: The Battle of Europe) In: *CIO Sweden 2008:03-05*.

Preece J., Rogers Y. & Sharp H. (2002). "Interaction Design: Beyond Human-computer Interaction". John Wiley & Sons, Inc: New York, NY.

Schwaber K. & Beedle M. (2001). "Agile Software Development with Scrum". Prentice-Hall: Englewood Cliffs, NJ.

Schwaber K. (2004) "Agile Project Management with Scrum". Redmond, Washington: Microsoft Press.

Swedish Trade (Electronic) "India" Available:
<<http://www.swedishtrade.se/indien/>> (2008-05-22).

Appendix

Appendix A: *Interview template.* Structured interviews with Software Engineers.

<i>Profile information</i>	
Image no.	
Name:	
Nickname:	
Family:	
Spoken languages:	
Hometown:	
Interests:	
Favorite place in India:	
Favorite movie:	
<i>Career</i>	
University education:	
Work experience (years):	
Programming work experience:	
All previous employers:	
Favorite/best programming language:	
Time at Indpro:	
Thoughts about Indpro:	
Improvements needed at Indpro:	
Career goal:	
Best work quality:	
Your role at Indpro:	

Current project:	
How has the implementation of SCRUM in your current project gone: (good/bad and lessons)	

Appendix B: Interview template. Projects specific and structured interviews with Software Engineers.

<i>General information</i>	
Project name:	
Client:	
Resources: (Developers and Managers):	
Abstract : (very short - purpose of project)	
<i>Client</i>	
Brief Client Description: (business)	
<i>Technical Overview</i>	
Language/s:	
Environment/s:	
Frameworks:	
Documentation/Testing:	
<i>Methodology & Execution</i>	
Method/s: (ex Scrum)	
Routines: (ex. meetings)	

<p>Communication/Reports:</p>	
<p><i>Status</i></p>	
<p>Project State:</p>	
<p>Developer Assignments – View of project: (process, handling, management, developer role)</p>	
<p>Project Problems (Identified – Handled – Resolved?)</p>	

Appendix C: *Interview template.* Semi structured interview with client.

Stage 1 [Present]

1. Current communication between team members in Sweden and India?
2. Differences between before 1st visit, after 1st visit, during 2nd visit.
Grounding communication.
3. What do you expect from Indpro: infrastructure/project leading/supervision.
4. What CMC tools do your company use for communication?
5. How did you choose the communication methods/tools to work with? Any regrets?

Stage 2 [Passed]

6. Changes made in Scrum methodology after distributing to India.
7. What [in your opinion] will improve the situation offshore/onshore?
8. What pit-holes have your company/you fallen into with the aspect on communication with India. Important lessons learnt.

Stage 3 [Future]

9. Prioritized future changes/investments.
10. Changing tools?
11. Key words to make it work.

Appendix D: *Ten rules of thumb for better usability.* Template used in heuristic evaluation

1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2. Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

3. User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. Recognize rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

TRITA-CSC-E 2008:117
ISRN-KTH/CSC/E--08/117--SE
ISSN-1653-5715