

*Master Thesis*  
*Software Engineering*  
*Thesis no: MSE-2009-28*  
*Nov. 2009*



# **Communication Support to Scrum Methodology in Offshore Development**

## **Case Study – Mermaid Technology, Denmark**

**Kashif Ali Sulemani,**

**Muhammad Nadeem Nasir**

School of Computing  
Blekinge Institute of Technology  
Box 520  
SE – 372 25 Ronneby  
Sweden

Internet : [www.bth.se/tek](http://www.bth.se/tek)  
Phone : +46 457 38 50 00  
Fax : + 46 457 271 25

This thesis is submitted to the School of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20x2 weeks of full time studies.

**Contact Information:**

Author(s):

Kashif Ali Sulemani,  
Address: Folkparksvagen 14 LGH: 11, 372 40, Ronneby, Sweden  
E-mail: kashifsulemani@gmail.com.

Muhammad Nadeem Nasir,  
Address: Folkparksvagen 14 LGH: 11, 372 40, Ronneby, Sweden,  
E-mail: nadeem.nasir@live.com.

University advisors:

Darja Smite,  
Senior Researcher / Assistant Professor,  
School of Computing,  
Blekinge Institute of Technology.

School of Computing  
Blekinge Institute of Technology  
Box 520  
SE – 372 25 Ronneby  
Sweden

Internet : [www.bth.se/tek](http://www.bth.se/tek)  
Phone : +46 457 38 50 00  
Fax : + 46 457 271 25

## ABSTRACT

In today world, the software companies are expanded above the continents. The software development work span continents and the distributed team work together over the borders. The agile development methodology demands close collaboration with client, rapid requirements change and an iterative development of fixed length. When this way of agile approach is applied in a distributed project, it requires frequent communication and knowledge exchange among the dispersed team members and need collaboration with customer over distance.

Besides the geographical, the linguistic and the different time zone barriers in a distributed project, the computer mediated tools suppose being useful media in connecting and to coordinating among dispersed colleague in a project. Though, these tools depict variant effectiveness in communication exchange, however, there efficient use connects peoples at the two sites. The aim of this thesis study is to explore the communication channel support to the Scrum practices in a distributed project. The purpose of the research is to analyse collaboration and communication in distributed teams working together through computer mediated technology. For this purpose, a company case is studied doing a distributed agile project. The author's insight the communication and information exchange through the ICT in an agile project. Based on the case analysis, the authors suggest the recommendations for implementing and establishing agile practices in a distance project.

**Keywords:** Distributed agile development, communication channels in offshore development, Distributed Scrum

## **ACKNOWLEDGEMENTS**

First of all we would like to thank Almighty Allah for blessing us with the abilities to do this thesis work.

We would also like to thank our father, mother, sister, brothers and other family members for their encouragement, moral support and dedication to pursue higher education in Sweden. Indeed, without their prayers, true love and moral support it could not be possible.

We are obligated to our university advisor, Darja Smite for her support, encouragement and supervision throughout the thesis work. Her valuable knowledge, timely feedback, and assistance on the thesis subject enable us to complete this work.

We would like to thank Mermaid Technology, Denmark for giving time. It could not be possible without their participation and support to complete this case study.

15 October 2009.

# TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	BACKGROUND .....	2
1.2	AIMS AND OBJECTIVE.....	3
1.3	RESEARCH QUESTIONS.....	4
1.4	EXPECTED OUTCOMES.....	4
1.5	RESEARCH METHODOLOGY .....	4
1.5.1	<i>Literature Review</i> .....	4
1.5.2	<i>Empirical Study</i> .....	5
<b>2</b>	<b>PROBLEM AREA DESCRIPTION.....</b>	<b>8</b>
2.1	AN OVERVIEW OF OFFSHORE DEVELOPMENT.....	8
2.1.1	<i>Main Characteristics of offshore development</i> .....	9
2.2	<i>An overview of agile development methodology – Scrum</i> .....	11
2.2.1	<i>Scrum Roles</i> .....	14
2.2.2	<i>Scrum Documents</i> .....	14
2.2.3	<i>Scrum Meetings</i> .....	15
2.3	UNDERSTANDING SCRUM PRACTICES IN OFFSHORE DEVELOPMENT.....	16
<b>3</b>	<b>COMMUNICATION CHANNELS AND KNOWLEDGE EXCHANGE – A CASE STUDY.....</b>	<b>18</b>
3.1	CASE DESCRIPTION.....	18
3.1.1	<i>The Case Company</i> .....	18
3.1.2	<i>The Product</i> .....	18
3.1.3	<i>Customer / Client</i> .....	19
3.1.4	<i>Distributed Team Classification</i> .....	19
3.2	SCRUM IMPLEMENTATION.....	20
3.2.1	<i>Roles</i> .....	20
3.2.2	<i>Documentation</i> .....	20
3.2.3	<i>Meetings</i> .....	21
3.3	COMMUNICATION SUPPORT.....	22
3.3.1	<i>Voice Communication</i> .....	22
3.3.2	<i>Instant Messenger – IM</i> .....	24
3.3.3	<i>Email</i> .....	26
3.4	KNOWLEDGE EXCHANGE SUPPORT.....	27
3.4.1	<i>Collaborative Tool</i> .....	27
3.4.2	<i>Team Discussion</i> .....	29
3.4.3	<i>Bug list</i> .....	29
3.5	CUSTOMER INVOLVEMENT IN THE PROJECT.....	30
	SUMMARY.....	31
<b>4</b>	<b>EVALUATION OF THE AGILITY LEVEL.....</b>	<b>32</b>
4.1	NOKIA TEST.....	32
4.2	METHODOLOGY.....	32
4.2.1	<i>Survey Study</i> .....	32
4.2.2	<i>Data Collection Method</i> .....	33
4.2.3	<i>Survey Participants</i> .....	33
4.2.4	<i>Nokia Test - Questionnaire Composition</i> .....	33
4.3	TEST RESULT.....	34
4.4	ANALYSIS AND DISCUSSION.....	35
	SUMMARY.....	38
<b>5</b>	<b>DISCUSSION.....</b>	<b>39</b>
5.1	APPLICABILITY OF THE SCRUM PRACTICES IN A DISTRIBUTED PROJECT.....	39
5.2	TOOLS-MEDIATED-COMMUNICATION AS A REPLACEMENT OF THE COLLOCATED COMMUNICATION.....	41
<b>6</b>	<b>VALIDATION AND THE VALIDITY THREATS.....</b>	<b>45</b>

6.1	VALIDATION OF THE RESULT .....	45
6.2	VALIDITY THREATS .....	46
6.2.1	<i>External Validity threats</i> .....	46
6.2.2	<i>Internal Validity Threats</i> .....	46
6.2.3	<i>Qualitative Validity Threats</i> .....	47
<b>7</b>	<b>RECOMMENDATIONS .....</b>	<b>48</b>
7.1	ESTABLISHING DISTRIBUTED SCRUM PROJECT SETUP.....	48
7.1.1	<i>The Agile methodology - Scrum Practices</i> .....	49
7.1.2	<i>Distributed Software Development Practices</i> .....	51
<b>8</b>	<b>EPILOGUE.....</b>	<b>53</b>
8.1	CONCLUSION .....	53
8.2	FUTURE WORK.....	53
<b>9</b>	<b>REFERENCES .....</b>	<b>55</b>
	<b>APPENDIX A: INTERVIEW QUESTIONS.....</b>	<b>59</b>
	<b>APPENDIX B – THE SURVEY QUESTIONNAIRES .....</b>	<b>60</b>
	<b>APPENDIX C – NOKIA TEST SCORE .....</b>	<b>62</b>
	<b>APPENDIX D – SURVEY RESULTS.....</b>	<b>64</b>

## **LIST OF TABLES**

<b>[Table 1.1]</b>	:	The person participated in the case study
<b>[Table 2.1]</b>	:	Scrum Concepts and Attributes
<b>[Table 2.2]</b>	:	Challenges of Offshore Development and Prerequisites for Implementing Scrum
<b>[Table 3.1]</b>	:	Scrums Meetings and Mode of Communication
<b>[Table 4.1]</b>	:	The Personal Participate in the Survey Study
<b>[Table 4.2]</b>	:	Team Scrum Practices along with their Nokia Test Score.
<b>[Table 4.3]</b>	:	Scrum Implementation and Revenue Earn
<b>[Table 5.1]</b>	:	Strategies to Minimize the Challenges in Distributed Agile Development
<b>[Table 5]</b>	:	The Communication Effective in Collocated and Distributed Project
<b>[Table 6.1]</b>	:	The team Scrum practices
<b>[Table 7.1]</b>	:	Distributed Agile Project Setup

## LIST OF FIGURES

- [Figure 1.1]** : Overview of Research Methodology
- [Figure 2.2]** : The Agile Manifesto
- [Figure 2.3]** : Scrum Work Process
- [Figure 2.1]** : The Distance Effects
- [Figure 3.1]** : The classification of team members at the onsite and offsite location
- [Figure 3.2]** : The software development activities in the study project
- [Figure 3.3]** : The Information Exchange in a Agile Project
- [Figure 4.1]** : The Team Score at the Nokia Test
- [Figure 4.3]** : Scrum Implementation at the Studied Company vs. Standard Scrum Practices
- [Figure 4]** : The Agility level Evaluation
- [Figure 5]** : The Case Discussion
- [Figure 7.1]** : The Distributed Scrum Project Setup

## Table of Abbreviations

<b>GSD</b>	Global software development
<b>SD</b>	Software development
<b>GSD factors</b>	Geographical separation, culture and linguistics dissimilarities.
<b>CMC</b>	Computer mediated tools
<b>ICT</b>	Information and communication tools
<b>IM</b>	Instant messenger

# 1 INTRODUCTION

During the last decade offshore software development has become a norm [1, 2]. In offshore software development organization relocate development activities into another country with their own [2]. As offshore development grows it brings unique challenges and benefits for researchers and organizations. The main drivers behind offshore software developments are cost, access to skilled labor, time shifting and market access [2,3,4,5]. However, its major issues are the far away working colleagues resulting in breakable information exchange, cultural and linguistic diversity [6,7 and 8].

Because of these challenges faced by offshore software development success rate is low as compared to co-located development. The Nisar et al. [2] mentions that in offshore development “7 out of 10 projects fail in some respect” [2]. The Herbsleb et al. [9] conclude that the distributed project estimate to required 1.5 more time when compared to co-located project duration. The Hazrati [10] conduct a survey to find out the success rate of co-located and distributed teams that are using agile methods for software development. It shows that success ratio of co-located team is 83%, distributed but physically accessible has 72% and the distributed geographically project has 60% success ratio.

Many other studies show that communication and knowledge exchange are one of the major issues in offshore development with agile methods [1,2,6,11]. In offshore project teams members are far away from one another and hence it is not possible to have face-to-face meetings on a regular basis [11]. They have to rely on computer mediated tools to communicate and to develop coordination with each other. As many industry practitioners conclude that the computer mediated tools are the only instrument has a valuable role in offshore development.

The focus of this study is on communication and knowledge exchange. Moreover the aim of this study is to explore the communication and knowledge exchange support that is required by agile method Scrum and design the recommendation for rich communication and knowledge exchange in offshore development with Scrum. For that authors have conducted a case study. For a case study an organization was selected which is doing offshore development with Scrum. This organization has main office in Denmark and the development centre in Pakistan. The offshore company is working on a software application for digital signage system. In this case study authors find out: the artifacts and communication channels necessary to maintain Scrum in offshore development, challenges and restrictions in maintaining Scrum communication, effective and least effective communication and knowledge exchange tools for Scrum offshore projects, and agility level in the studied organization with the help of “*Nokia Test for Scrum*”.

This study is divided into eight chapters. First chapter describe the background, aim, objectives and research methodology for this study. The second chapter explains the problem area description. The third chapter describes the case study in the organizations. The chapter four layouts the agility level in the studied organization. The chapter fifth provides overall discussion on the study. The sixth chapter provides the recommendations for communication and knowledge exchange in offshore development with Scrum. The chapter seven correspond the validation of the study result and the detail about the validity threats in the case project. And the last chapter finishes with the thesis conclusion and the future work area.

## 1.1 Background

Agile development method - Scrum is one of the growing development methods in software projects [13]. Scrum is a process skeleton that includes a set of practices and predefined roles [13, 14]. The Scrum team composed of Scrum master, Product owner and development team. A set of practices include Scrum sprint and Scrum meetings. Following Scrum meetings are part of project management: Daily Scrum meeting, Sprint planning meeting, Sprint Review meeting and Sprint Retrospective meeting [13, 14]. One of key success in agile methodology Scrum is collocated development, verbal communication and team collaboration [14].

The problem with agile method is when applying in offshore development [15]. In offshore development, distributed team members are far apart from each other. The challenges in managing distributed team members are poor communication and complex knowledge exchange [15]. Scrum is an agile method that demands a close collaboration between developers, customer and the project manager [13, 14]. This collaboration suffers when project members are far apart from each one another. In offshore development using Scrum, development part is usually done at offshore country and client resides in at another location. Client leads the offshore projects as the Product owner and the project manager as a Scrum master [14]. Another issue in adopting distributed Scrum is the exchange of knowledge among distributed teams [13,1]. The main Scrum artifacts include Product backlog and Sprint backlogs. The Product backlog is managed by Product owner whereas the Sprint backlog is property of development team and it includes the task descriptions. In a distributed project, development team at one geographical region requires the Product backlog to implement client requirements and the wish list. Whereas, the product owner and Scrum master at the other end needs to view the current Sprint status and task descriptions.

To connect the distributed team members, there is a need of an effective communication [17] which acts as a bridge in offshore development sites. The agile development method has characteristics of rapid development, an active project client participation in development process and self organized project team. A few case studies [16, 54] depicts that a least or rare information exchange among the project stakeholders result the complete failure of offshore project. Similarly, the others research [16] shows that discontinuous (or breakable) information exchange among the project team members slow down the development process resulting the development processes delayed on its schedule time. Besides this, the information exchange significance in an offshore project, one of the effects of least-communication is the 'lack of trust' among the project stakeholder [38]. In short, communication is of great importance, that if it is not taken seriously in an offshore project, all practices of Scrum method like daily Scrum meetings and Scrum rapid development will be lost. In offshore development, communication is a way to link together Scrum actors (Scrum Master, Product owner and Scrum team). It is reported to reduce geographical distance in offshore project [18] and help to develop collaborative team [17, 18]. Similarly the use of collaborative tools makes possible knowledge sharing among Scrum team members [17, 18]. Collaborative tools act as a central repository of Scrum data in offshore development [17]. It enables Scrum roles to access the Scrum artifacts despite geographical distance. The communication channels and knowledge transfer tools have significant importance in offshore Scrum projects [18]. The unavailability of each of these has adverse effect on the rapid software development.

The other alternative way to support Scrum activities (e.g. Scrum meetings, Scrum Sprints) is face-to-face communication [14]. Due to geographical distribution of the team, travel time and travelling expenses, it is not feasible to organize face-to-face meetings often in offshore development [14, 15].

From the above arguments it is concluded that communication and collaboration tools play significant role in geographically distributed teamwork. Communication management is the key element in offshore project planning and coordination. Different synchronous and asynchronous tools help Scrum team to coordinate with each other. Similarly the collaborative tools like groupware help to share Scrum artifacts among distributed team members [18, 19].

The purpose of this study is to explore the communication support that is required by an agile method Scrum in offshore development. The intent of this study is to identify the collaborative and communication tools needed to support Scrum method in different stages of an offshore project and investigate their applicability and effectiveness in the context of the distributed agile project. The study will help industry practitioners to understand the issues before implementing the agile methodology Scrum in distributed project. The identification of right communication channels and knowledge transfer tools will be a little contribution to current literature knowledge of 'Offshore Scrum development' especially considering that these are based on empirically validated observations.

The current literatures address the communication challenges and its support for distributed software projects [15, 1, 18]. But it rarely addresses its ultimate need for agile software development methodologies. This thesis is an effort towards contribution to address the communication support provided to Scrum method in offshore development.

The authors of this study used literature study and empirical study as a research methodology. During this research the authors investigated about the Scrum members' roles, Scrum meetings and the Scrum sprint. It was explored how communication provides support to these Scrum elements in distributed projects and identifies communication channels and collaborative tools to support Scrum methodology in offshore development. In addition the effects of team members distribution into two continents was investigated. Further it was examined how communication between two sites suffers due to time zone differences and language barriers in Scrum projects.

Although Scrum methodology is designed for in-house development [13, 14, 19], we believe that it can be applied and managed in offshore development as well. The authors did a research to investigate the efficient use of communication channels to support Scrum methodology in offshore software projects.

## **1.2 Aims and objective**

The aim of this thesis is to explore the communication and knowledge exchange support that is required by agile method Scrum and the ways of offshore development. For this the following objectives are required to be fulfilled.

- Identify restrictions of applying Scrum in offshore development
- Identify communication channels including Scrum roles and activities that require communication support in offshore development.
- Identify communication and knowledge exchange tools, which provide rich support to Scrum project and their application in industry.

- Categorize the Scrum supported communication channels and knowledge transfer tools into effective and least effective ones considering time zone differences and language barriers.
- Find out the agility level of a Scrum project in offshore development
- Propose recommendations for building Scrum project infrastructure in offshore project.

### **1.3 Research Questions**

Following are research questions that need to be addressed during the thesis:

- RQ1.** What are the restrictions caused by geographic and temporal distribution and language barriers, for maintaining Scrum communication and knowledge exchange in a distributed project?
- RQ2.** What communication and knowledge exchange support does the agile methodology Scrum requires?
- RQ3.** What are the communication channels and knowledge exchange tools used in industry? Which tools are effective, which are ineffective?
- RQ4.** What is the agility level of project team?
- RQ5.** What are the recommendations to implement the Scrum practices in a distributed project?

### **1.4 Expected outcomes**

The expected outcomes of this study are related to communication and knowledge exchange support in offshore development using Scrum. Thesis report contains the following outcomes:

- List of artifacts and communication channels necessary to maintain Scrum in offshore development.
- List of challenges and restrictions in maintaining Scrum communication.
- The evidence of industrial practices concerning communication and knowledge exchange tools practically used in a Scrum offshore project.
- Observations of effective and least effective communication and knowledge exchange tools for Scrum offshore projects.
- Evaluation of the average of Scrum practices in a Scrum offshore project.
- Recommendations for establishing Scrum project infrastructure in offshore development.

### **1.5 Research methodology**

A mixed methodology approach was used for this research study. It evolved the qualitative study by doing a literature review on the challenges and the issues of agile development methodology in distributed environment. And the other part of study includes the empirical research, which composed of a semi structure interviews and a survey study to investigate the applicability of an agile practices in a case project. These two research approaches are described as follows:

#### **1.5.1 Literature Review**

A literature review was conducted dealing with two major areas of research. These are:

**1. An agile development methodology – Scrum.** It involved the Scrum roles, artifacts and practices. In this part, the literature study were carried out about the Scrum roles in the software development; the iterative development practices and the documents/artifacts needed or produced during agile way of development. In this literature study, the authors focus was to identify the prerequisite of the Scrum approach and to identify the problems in maintaining it. The formulation of this part of literature study is given in the Chapter 2.

**2. A literature review of various GSD challenges** – This involved a literature review of challenges and issues of GSD. In the literature study, the challenges and issues of GSD focused were the geographical separation, culture and linguistic dissimilarity. The description about this part of literature study is given in the Chapter 2.

The literature study analysis was to investigate the challenges and issues counter in adopting the agile practices in a distributed project.

### 1.5.2 Empirical Study

The empirical research was undertaken by studying the agile distributed case project In the studied project the data gathering and analysis methods included: interviews with the remote and onsite team members, face-to-face discussions with the software developers and by studying project manuals and documents. In this section, we will discuss about the studied project, the study group, the questionnaire composition and the ways of collecting data.

- **A Case Project – Mermaid Technology**

A Mermaid technology is a Danish based company specialized in manufacturing, installation and working of ‘digital signage system’. The company has the software development part of their business to the Asia – Pakistan region. The company has been practicing global software development for the last five years. The authors only focused on the distributed way of software development. The software development methodology in the studied project was the agile approach where the both offsite and the onsite team members actively participate in the software development processes. The reason to choose the Mermaid technology as a study project was that the reference company followed the agile way of working and the project span over the two countries having different cultures and did not share the same time zone.

- **Study Group:**

In order to order to explore the distributed agile development and the communication exchange in the studied project, the authors contacted the peoples actively involved in the case project. In this case study, the authors have classified the study participants into the following three groups; the development team, the management team and the Danish client. The purpose of the member’s classification is to get an accurate point of view from each group and to explore the problems faced when communicating with each others. The development team participants include the team leads, the software developers and the testers personals reside at the remote site. The management team participants include the project managers, the system architecture and the requirement engineer. The total numbers of persons in participatory study were 14. The table 4.1 depicts the study groups and the members participate in the study.

**Table 1.1. The person participated in the case study**

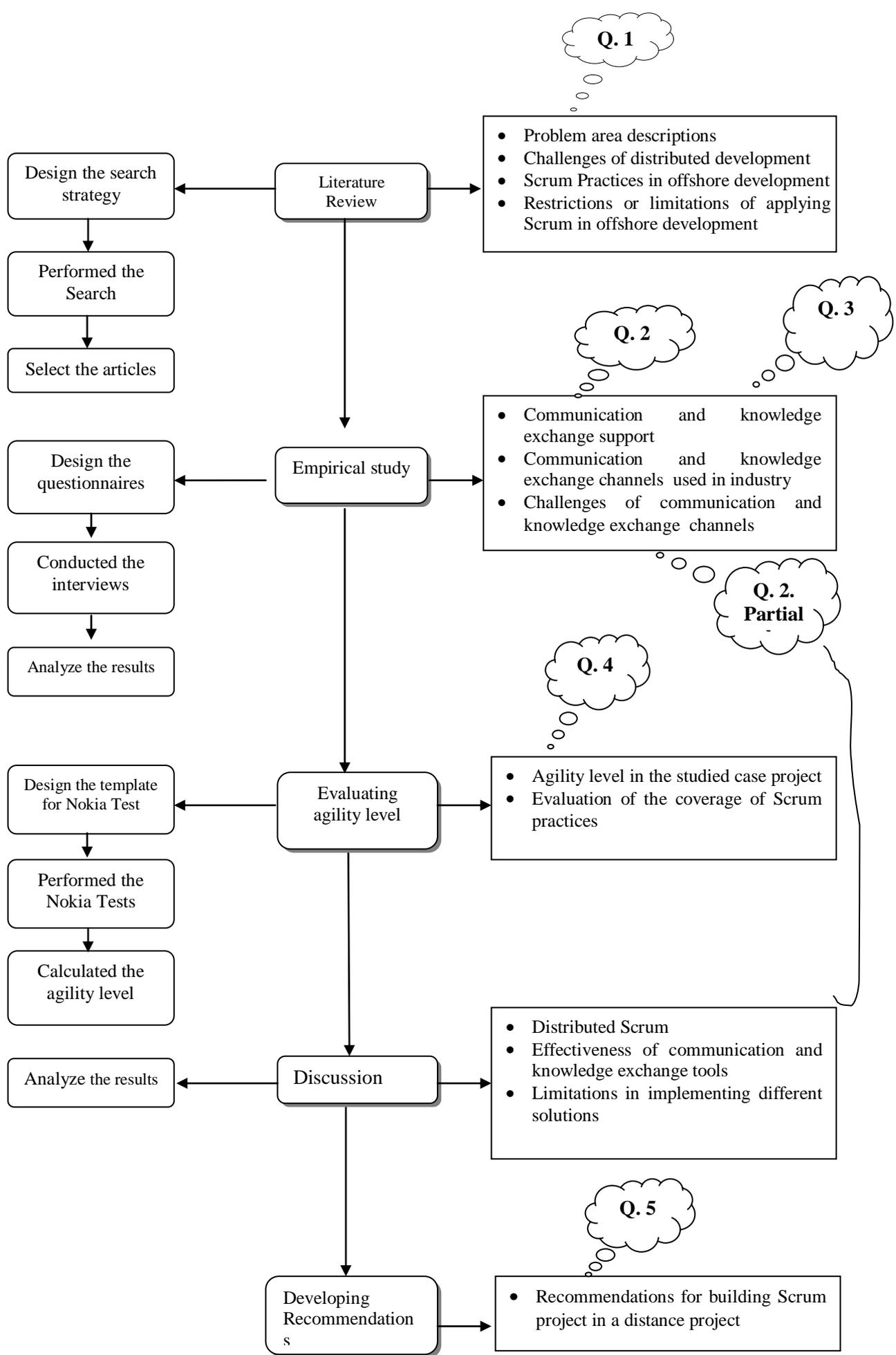
<b>Development team members Offsite team</b>		<b>Management team members On-site team</b>	
Team leader/Senior developer	2	Project manager	2
Software developers	5	System designer / Flash designer	2
Testers	2	Requirement engineer	1
Subtotal	9	Subtotal	5
<b>Total = 14</b>			

- **Data Collection Methods**

The research in the case study was conducted by semi structure interviews, by studying project documents and manuals and by conducting a survey study among the participatory members of the studied project. The main aim of these data collection techniques were to examine the iterative way of development, to study experiences, to identify the issues and to examine the GSD factors influence on the distributed agile work.

The authors have conducted nine - 9 interviews in total from the onsite and the offsite team members of the project. In these nine interviews, seven – 7 interviews has been conducted from the offsite team members and the three -3 interviews has been conducted from the Danish site. The interviews questions had been send in advance through an email. Similarly, the appointment and the exact time for the Skype communication have been taken through an email. We asked them for one hour interviews and tried to ask as many questions as possible in the interview session. Among these interviews, the authors get an opportunity to conduct two of the interviews face-to-face which last for an hour. The interviews answers were documented to reach the final result of our focus area.

Besides the interviews from the study group, a survey study was conducted in order to perform the Nokia test in the studied project. The purpose of the Nokia test was to evaluate the agility level of the distributed Scrum team. The members participated in the survey study are shown in table 1.1. A survey study was performed in mainly two ways. By sending the survey questionnaires through an email and also through the online survey portal. The formulation of survey questionnaires can be found at Appendix B of this report.



**Figure 1.1: Overview of Research Methodology**

## 2 PROBLEM AREA DESCRIPTION

### 2.1 An Overview of Offshore Development

Offshore refers to the place where the work tasks are performed. It's the term that applied when the work is sent over another country [2, 24]. In offshoring, organizations relocate development activities abroad with their own affiliation [25, 2]. Offshoring is different from outsourcing. In outsourcing organizations carries out development activities to another organization that are located either locally or in another country [25, 26].

In 90s, there were significant number of jobs in IT sector but less skilled labor in US, especially during fixing Y2K problems and creating new internet products and services during the dot-com boom [24]. In order to fill this gap, organizations started to relocate their work abroad, because of insufficient US labor market. This phenomenon opens a new horizon for organizations. Organizations started to offshore their business processes to countries which have skilled labor, favorable tax policies and low wages. With the passage of time offshore increased and it offered many opportunities and challenges for individuals, organizations and policymakers [24]. Offshoring gained popularity in both developed and developing countries and now it is more popular than ever before [3, 4].

The primary driving force behind offshoring is cost [5]. Organizations which want to reduce the development cost, started offshoring in countries which give them this opportunity. Beside this, some other reasons behind offshoring are: access to skilled labor [4], experience, time shifting, time to market, market access [4], extending venture capital money [24], and increased team flexibility and ability [4]. Although offshoring has advantages and benefits, there are few challenges faced by offshore development like: geographical distance [38,39], temporal distance[38,39], cultural distance[38,39], linguistic barrier[38,39], poor communication and knowledge exchange [1,11,33]. Study shows that seven out of ten projects fails in offshore development because of the reasons mentioned above [2, 27]. These challenges are some significant hurdles in achieving full benefits of offshore development.

In today world, it's a fact that offshore development projects are in large size, and involves people working at different locations [11]. As number of people working on a project increases, the level of software development cooperation also increases. New methodologies and skills are required to handle the team at different locations which have different background, knowledge and skills.

Offshore software development demands emerging development approaches which increases productivity, performance and reduce the failure rates. These approaches are different from in-house development which meets rapid change in requirements and demands frequent communication between development team.

Agile development methodology is one of the approaches that address the above mentioned challenges. It has been applied in co-located projects. The agile method Scrum was proposed to attain the hyperproductive state where "*productivity increase 5-10 times over industry averages*" [13]. This hyperproductive state is achieved by many teams in co-located software development [13]. The Sutherland *et al.* [13] conducted a case study in Xebia that started project with Dutch and Indian team members. The theme of this case study is whether distributed offshore teams can

always achieve the hyperproductive state. This study concludes that it is possible to obtain the same velocity and quality like in co-located project. This paper discusses the model that Xebia productized after using XP practices within many distributed Scrum projects. This model is about high performance, distributed offshore development team and high quality [13].

The Schwaber [29] describe the Scrum development process in detail. He also gives a comparison of Scrum with traditional development methodology (waterfall, Spiral and Iterative). He point out that main difference between waterfall, Spiral, Iterative and empirical Scrum methodology is that the Scrum methodology assumes that the analysis, design and development process during the Sprint is unpredictable. A control mechanism is used to control the unpredictability during the Sprint and as a result it provides flexibility, reliability and responsiveness [29].

The Lucas *et al.* [11] presented a case study on essential communication in a globally distributed XP team. This case study provides challenges faced by the project management team in USA and the software development team in the Czech Republic. The main goal of this case study is to understand how this globally distributed team develops a successful project in a new problem domain using a method that is reliant on face-to-face communication. They suggest four success factors for globally distributed XP team [11].

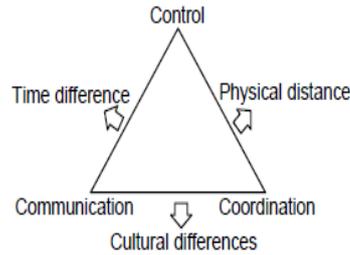
The Dybå *et al.* [23] conducted a systematic review on empirical studies of agile software development in 2008. In this systematic review 1996 studies are search out from different sources and 36 studies are found acceptable. The studies are categorized into introduction and adoption, human and social factors, perceptions of agile methods and comparative studies. They write down the number of potential benefits and limitation of agile development. The clear finding of this systematic review is that there is need to increase both the number and quality of research articles on agile software development. This study particular mention that Scrum method required further attentions [23].

The French *et al.* [16] presented a study on communication and cooperation in distributed software project teams in five organizations. This study mentions that as software development and maintenance always involved a number of stakeholders like: programmer, testers, designers and end users therefore need of communication and cooperation increase. They highlights the alternative need of communication and cooperation in distributed software development and provides lesson learned from this study.

From the above arguments it is observe that Scrum (an agile development methodology) is a process that requires an informal, rich communication environment in order to succeed. Communication and knowledge exchange is one of the central values of Scrum. Many of Scrum role (Product owner, Scrum Team, Scrum Master) and Scrum meetings (Sprint planning, Sprint review, Daily Scrum) require rich communication. Scrum documents (Product backlog, Sprint backlog, Burn down chart) require knowledge exchange between stakeholders.

### **2.1.1 Main Characteristics of offshore development**

The distributed software development teams face challenges in communication, coordination and controlling software development because of three reasons, these are physical distance, time difference and cultural difference as shown in figure 2.3.



**Figure 2.1: Distance effects: Source [33]**

- **Physical Distance**

Due to physical distance it is difficult to make intense communication and knowledge exchange across sites for various reasons like: Face-to-face communication is not possible, developers at one site are unable to share same environment and they do not know what is happening at the other site, developers at a single location meet with each other, these frequent meetings are not possible due to physical distance [33].

- **Time difference**

If time zone difference between business centers and offshore development sites is significant then there are very few hours in the day when projects participants of both sites are available in the office [2]. This makes it essential to depend upon asynchronous communication which is not as effective as face to face communication and causes delay in explanations and clarifications to the next working day [33].

- **Culture difference**

The cultural difference is a major factor and it cannot be neglected when doing an offshore development [2, 33]. Depending upon the offshore geographical location, the cultural differences vary from minor if its nearshore country, for example an offshore centres in Eastern Europe countries. However, there is a much cultural mismatch if a project offshore to Asia such as India or Far East Asian country. So far India has been the most favoured outsourcing destination [19, 37]. A common cultural attributes possess by the Indian people at work [67] are as under.

The Indian peoples are never comfortable saying “No”, to anyone even to their project customer. However, in software business, the other culture peoples will not mind just because the remote member is unable to do or just say “No” to the project customer.

Similarly, the mostly Indian peoples do not feel comfortable giving bad news. The people have believe that they should always give a good new to others. Further, when it comes to giving the bad news, the peoples find alternative ways to convey the bad news. It is an important culture point to understand, since many project customers get “surprises” about their project related issues disclosed at a very later stage and become wondering why the issue wasn’t came into consideration during the meeting.

## 2.1.2 Main Challenges of offshore development

- **Communication Challenges**

In offshore software development two teams that have different technical backgrounds, different approach to work, located at different time zones, joined together to deliver a project, this requires coordination, communication, knowledge transfer and ability to revise work on regular basis [33]. These demand trust and mutual understanding among team members so [36] say that communication among team members plays an important role in the success of a distributed project. Communication particularly is imperative for Scrum team due to its nature and structure. Communication is fundamental factor and is indispensable for smooth management of team, project and to write fast and high quality code.

- **Knowledge Sharing Challenges**

With effective information and faultless knowledge exchange mechanism, management can exploit the benefits of offshore software development [37]. New challenges are introduced when managers might be unable on timely and regular share information with stakeholders and team is unaware of critical tasks. In offshore software development besides effective information some other artifacts like poor documentation, updating and revising the documentation is equally significant. To avoid assumptions, ambiguity and to hold maintainability, documents must be updated. That shows what different teams are using and developing on.

- **Human Challenges**

In offshore development, teams are distanced some thousands of miles, some time zones apart and strive for common goals. Each team has an area of specialization and they have different knowledge, background, environment and lack of visibility. The outcome of the offshore software development depends upon team ability and it is difficult to develop trust between remote team members [27, 34].

- **Technical Challenges**

Since computer networks are expanding globally and different countries are using different standards for networks [34, 37], so it is possible that dispersed sites are slow and unreliable. The technical challenges also arise due to infrastructure issues at different sites [34,37]. This would result in unproductive meetings and sometimes the meetings need to be rescheduled or abandoned. IT infrastructure plays a significant role in the success of offshore software development because it is compulsory part of offshore development. [33,55]. IT infrastructure needs to be secure, reliable, and simple.

## **2.2 An overview of agile development methodology – Scrum**

There are several ways of how to organize and work in software development projects. One way is to use an agile development methodology that provides a framework for all phases of the software development. The Agile Manifesto (Figure 2.1) describes the principles followed by agile software development and is based upon four statements. The Agile Manifesto highlights the importance of interaction, collaboration, dynamic way of working. It is observed that all stakeholders in the project are actively involved in software development planning and progress.

### Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

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

That is, while there is value in the items on

Figure 2.2: The Agile Manifesto, (Source [28])

Agile framework was developed in mid 1990s [28]. There are many flexible method develop after and before agile framework. Some of these methods are: Scrum, Extreme programming (XP), Test driven development, and lean development.

The term “scrum” came from Rugby game, it is the approach that brings an out-off-play ball back into play. The Rugby and Scrum team is adaptive, quick, and self-organizing [29]. The Scrum method was developed by Jeff Sutherland and Ken Schwab in 1996 [19]. Scrum is an agile method and most popular in industry [30]. Scrum is basically focused on what is to be developed; it has set of guidelines and no strict rules [30].

The Scrum works with objectives, priority requirements as compared to traditional software development with predefined requirements where goal is to fulfill the requirements [30]. Scrum classifies number of roles, documents and meetings, as shown in figure 2.2.

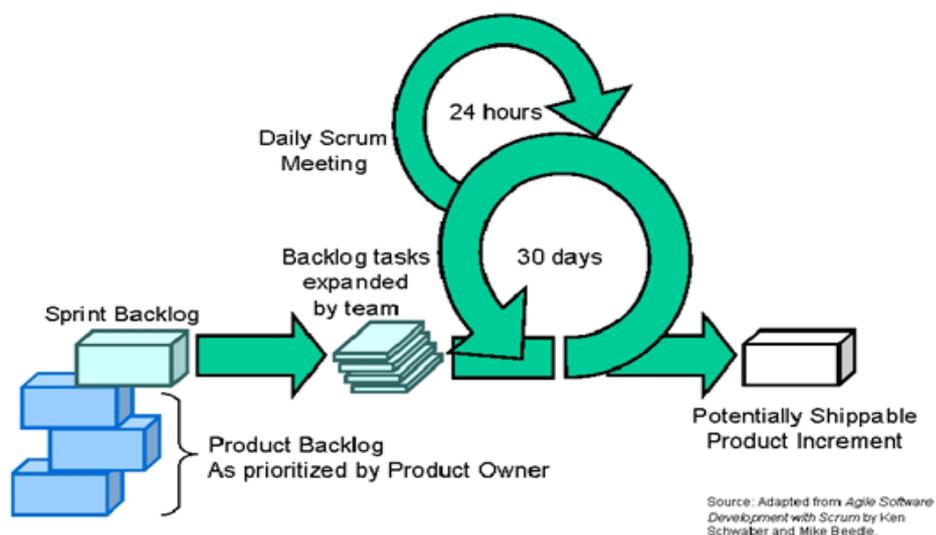


Figure 2.3: Scrum work process, (Sources [30, 31])

The Table 2.1 provides an overview of Scrum methodology. This overview is created on the basis of literature review of articles [29, 30, 31, 32]. The table provides the list of roles, artifacts and meetings of Scrum.

**Table 2.1. Scrum concepts and attributes.**

<b>Concepts</b>	<b>Attributes</b>
Product Owner	<ul style="list-style-type: none"> <li>• Manage the Product backlog</li> <li>• Communicate and exchange knowledge with Scrum Master, developers and customers</li> <li>• Make sure that process is visible for all stakeholders</li> <li>• Prioritized the requirements with the help of team</li> </ul>
Team	<ul style="list-style-type: none"> <li>• Team consists of software developers, small team 5 to 9 members</li> <li>• Self organized team</li> <li>• Cross-functional</li> <li>• Independent</li> <li>• Communicates and exchanges knowledge easily</li> <li>• No imposed structure within team</li> </ul>
Scrum Master	<ul style="list-style-type: none"> <li>• Make sure that process is followed accurately</li> <li>• Create bridge between team and management</li> <li>• Share information about team progress</li> <li>• Keep track of team progress</li> <li>• Remove handles</li> <li>• Provide required resources, independent to team</li> </ul>
Product Backlog	<ul style="list-style-type: none"> <li>• Contains work task for Scrum project, description of features, items and priority of items. The task/sprint estimates calculation by planning poker by collaborative stakeholders of a project.</li> <li>• White boards, excels sheets might be used for planning poker and to calculate task/sprint/project estimation in the Product backlog</li> </ul>
Sprint Backlog	<ul style="list-style-type: none"> <li>• Contains task line for current sprint</li> <li>• Software development teams share current Sprint status, task description with Product owner, Scrum Master and developers.</li> </ul>
Burn down chart	<ul style="list-style-type: none"> <li>• A Task/Sprint/Project velocity</li> <li>• It is used to show remaining work versus time</li> <li>• A graphical representation about the status of task/sprint/project</li> </ul>
Sprint Planning Meeting	<ul style="list-style-type: none"> <li>• A face to face meeting between team, customers and management</li> <li>• Bring goal of next sprint</li> <li>• Required communication and knowledge exchange in order to succeed.</li> </ul>
Daily Scrum	<ul style="list-style-type: none"> <li>• Face to face meeting</li> <li>• Short meeting near about 15 min on daily basis</li> <li>• Every team member answers there questions</li> <li>• Not documented</li> </ul>
Sprint review meeting	<ul style="list-style-type: none"> <li>• Face to face meeting</li> <li>• Provide review of what they have build during the Sprint</li> <li>• It is held between Product owner, management, and users</li> <li>• Share good stories , bad stories and experience with each others</li> </ul>

Next paragraphs explain all these Scrum roles, artifacts and meetings in detail with an emphasis on the challenges brought by distance.

## 2.2.1 Scrum Roles

Scrum has three roles for people working on it. These roles are Product Owner, Scrum Team and Scrum Master. Description of each role is give below.

- **Product Owner**

Product owner is a person which stands for the interests of stakeholders in the final system [30]. The Product owner maintains and controls the Product Backlog [31] and Product Backlog contains all the work tasks of Scrum project. He or she performs this duty with different names such as: project manager, product manager or anyone else who is responsible for the product. Product Owner prioritized the requirements in the Product Backlog. He keeps the Product Backlog manageable like priority assignment. No one can change the priority of Product Backlog without the permission of Product Owner. Product Owner takes help from the team in order to estimate how much time is required to implement an item or task.

- **Team**

The teams consist of software developers. The Scrum believes small, self-organizing, cross-functional, and independent teams [30]. Each team has five to nine people [31]. If you want to handle large number of people then divide them into multiples teams. Small teams have quite low opportunity of synergy effects through interactions and on the other hand large teams do not have time for self organization. Scrum does not offer any roles or imposed structure within team. The team is self-organized according to the situation. This will provide flexibility to team; adapt a changing environment, make easy communication and knowledge exchange.

- **Scrum Master**

The Scrum Master plays an important role in Scrum. He /she is the person who makes sure that Scrum process is followed accurately [30, 31]. The Scrum Master establishes a bridge between the team and management. Scrum Master gets information about the team member progress and what hurdles they are facing. It is Scrum Master responsibility is to remove hurdles so that team can continue with assigned work. Scrum Master also works with the management to make sure that the team has the required support, the resources, and independence that is necessary. Scrum Master keeps track of the team progress and informs others about work progress.

## 2.2.2 Scrum Documents

Scrum prescribes creation of the following documents: Product Backlog, Sprint Backlog, and Burn down chart. These documents are used to hold the work task for Scrum project. Following sections provide description of these documents.

- **Product Backlog**

A Product Backlog is the list of requirements needed to be done by the project team. It contains work tasks for the entire Scrum project [32] and Product Owner maintains the Product Backlog. It contains descriptions of necessary features, items

and priority of these items. It also contains estimated time in days which helps the Product Owner to measure the timeline and priority.

The product backlog list the requirements have a value for the customer. It includes both the functional and non functional requirements detail. It also includes the items required by the project team the ones that will bring value to the final product [29, 30]. The document does not typically include the detailed requirement information. Usually the requirement detail is figure out at the time of its implementation along consultation with the product customer. Similarly, the document does not include concrete low level tasks detail.

Besides this, the document utilizes the simplest and effective way of requirement prioritization, a wish list. The project customer along with consultation with the project manager and the development team marked the items priority and categories all of them together in each Scrum sprint.

- **Sprint backlog**

The Sprint backlog contains the task list that the Scrum team is going to build in the current Sprint [31]. The items in a sprint backlog are selected from the Product Backlog based on priority and time that it will take to complete the different features. The selection of item and size of Sprint backlog is very critical for Scrum team because they have to complete the selected item within time. After Sprint is locked, then changes should not be made in that Sprint. The Scrum team is free to work with and prioritized the items as they like.

For Sprint backlog Excel sheets, SharePoint, or defect tracking system are used. Usually same tools are used for Sprint backlog and Product Backlog. Sprint backlog is maintained by software development teams. They have to share current Sprint status and task descriptions with Product Owner and Scrum Master, they have to communicate with stakeholder for creation of Sprint backlog, development team have to update Sprint backlog on daily basis. These tasks require effective communication and knowledge exchange mechanism between stakeholders.

- **Burn down chart:**

Burn down chart represents remaining work versus time in graphical form [31]. The vertical axis shows the remaining work and horizontal axis represents time. This chart represents time when all the work will be finished. The Excel sheets, SharePoint, whiteboard are used to create Burn down chart; Project manager creates and maintains burn down chart with the help of team. Project manager needs remaining hours of task from development team on daily basis to update the burn down chart.

### 2.2.3 Scrum Meetings

The Scrum team normally holds Sprint planning, Daily Scrum and Sprint review meetings. The following sections provide brief overview of these meetings:

- **Sprint planning**

Scrum lifecycle starts from Sprint planning meetings [30]. In the first Sprint planning meeting, team meets with stakeholders such as: customers and management. This meeting brings out what to be build in the next sprint. In the first meeting, Product Owner gives presentation of prioritized items then holds an open discussion

that decides necessary changes in Product Backlog. The factors that are taken into consideration are: items delivered by the team at the end of Sprint, team capabilities, and technology stability. The team and other participants decide which items they believe they can build in the next Sprint. After that they decide Sprint goal. The Sprint goal is the objective of the Sprint. The Sprint goal is one sentence statement about Sprint. When the Sprint goal is identified the second Sprint planning meeting takes place. The second meeting is held between the members of Scrum team. The objective of this meeting is to come up with Sprint backlog [31].

This Sprint planning meeting requires communication with the team, Product Owner, customers and requires exchange of documents such as Product Backlog between all participants [31]. So build a Sprint backlog that is acceptable for all team members [31].

- **Daily Scrum**

Scrum team meets on a daily basis during each Sprint; this meeting is called daily Scrum which is short and concentrated. The duration of daily Scrum is usually 15 minutes. In this meeting every team member answers the following three questions: *What did you do yesterday? What will you do today? What obstacles are in your way?* [31]

- **Sprint review**

A Sprint review meeting is held when sprint comes to an end. In this meeting Scrum team presents what they have built during the Sprint. The Product Owner, management, and users participate in this meeting. The Scrum team notifies good and bad stories that held during the Sprint. The main focus is to demonstrate the Product. Normal duration of this meeting is about four hours. In this meeting development team shows their progress to the Product Owner, management, and customers.

## **2.3 Understanding Scrum Practices in Offshore Development**

Scrum roles, meetings and artifacts demand rich communication and knowledge exchange. This communication and knowledge exchange suffers when teams are dispersed at geographical boundaries. Scrum development faces some restriction because of geographical distance, temporal distance, cultural distance, and language difference when Scrum applies in offshore development. This section discusses limitations that apply on Scrum because of these challenges. The table 2.2 provides an overview of offshore development challenges and prerequisites for implementing Scrum.

The table is created on the basis of literature review. A literature review is conducted to analyze the challenges of offshore development and prerequisites for implementing Scrum and how to make offshore development projects successful with agile methods.

**Table 2.2. Challenges of offshore development and prerequisites for implementing Scrum**

<b>Offshore development Factors</b>	<b>Challenges of offshore development [39,40,41]</b>	<b>Prerequisites for implementing Scrum [39,40,41]</b>
Geographical distance	<ul style="list-style-type: none"> <li>• The team members virtualness</li> <li>• Invisible development process</li> <li>• Problems in communication and knowledge exchange</li> <li>• Limited face to face meeting</li> </ul>	<ul style="list-style-type: none"> <li>• Co-located, self-organized team</li> <li>• Visible process , Frequent iteration</li> <li>• Required everyone to communicate and share knowledge</li> <li>• Scrum focus on daily face to face meetings</li> </ul>
Temporal distance	<ul style="list-style-type: none"> <li>• Asynchronous communication</li> <li>• Problems solving hard and slow</li> <li>• Difficult in monitoring the work</li> </ul>	<ul style="list-style-type: none"> <li>• Synchronous communication</li> <li>• Fast feedback, quick response to customers requirements change</li> <li>• Symmetric work</li> </ul>
Cultural distance	<ul style="list-style-type: none"> <li>• Difficult to develop trust and relationship between team.</li> <li>• Misunderstanding increase because lack of awareness, interpretation and meaning of different terms.</li> </ul>	<ul style="list-style-type: none"> <li>• Scrum focus on team building, trust and relationship</li> <li>• Demand awareness, visibility of process and transparent project documentation</li> </ul>
Linguistic barrier	<ul style="list-style-type: none"> <li>• Unable to communicate with remote site, results in low communication.</li> <li>• Misunderstood, unclear and ambiguities in project documentation</li> </ul>	<ul style="list-style-type: none"> <li>• Required everyone to Frequently communicate and share knowledge</li> <li>• Transparent project documentation</li> </ul>

## **3 COMMUNICATION CHANNELS AND KNOWLEDGE EXCHANGE – A CASE STUDY**

### **3.1 Case Description**

A case study presented in this thesis focuses on communication and knowledge exchange support investigated in distributed project of Mermaid Technology, Denmark. The company is specialized in installation and IT related solution of digital signage system. The reason to choose the ‘Mermaid Technology’ is that the company has offshore IT solution part of their business to one of a remote site. In this section, the authors will discuss about the company, its communication with their remote partner, and methodological activities the company usually practises in its offshore project.

#### **3.1.1 The Case Company**

Mermaid Technology is a Denmark based company and was founded in 1990. The company is specialized in the manufacturing and installation of digital signage. The company provides hardware and software solutions to their respective clients/customers. The hardware components include TV screen, screen channels, amplifier, broadcaster system etc. The software service includes an application suite to broadcast advertisement or product commercial on these screens hoarding. Mermaid is one of the leading Signage Companies in North Europe. The other competitor of Mermaid was the Victor soft, Denmark. In 2003, Mermaid Technology purchased all the shares of Victor soft and became a leading company in digital signage system.

In 1998, the company has started designing the ‘digital signage’ application software. After few years of software development, the company has decided to offshore development part of their business to a developing country. The reason to switch towards offshoring was high employee wages, high tax rates and in some extent lack of technical persons in Denmark. Hence, in 2002, the company has signed an agreement with the Media Soft, Pakistan for the prototype development of ‘Digital Signage’ application. The Media Soft was a small size company located in Lahore, the IT hub of Pakistan. The Media Soft had many years experience in ‘Multimedia development’. The team of Media Soft, Pakistan was technically competent and was good in English language skills. These were the few reasons which motivated the top management of Mermaid Technology, Denmark to switch development part of their business to Pakistan.

#### **3.1.2 The Product**

The Mermaid Technology provides a vast range of hardware and software products. In this case study the author’s only focus is on the software application development. The software application suite designed by the company is named as ‘VTouch Pro’. VTouch Pro is a digital signage product for central advertisement. It is used for assembling rapid advertisement on the spot at different places. ‘VTouch Pro’ is

software which helps to send and receive information/advertisement from different locations within short time period. For example one of the company clients has 500 stores in the country, and at each store there are around 6 - 20 screens hoarding in it.

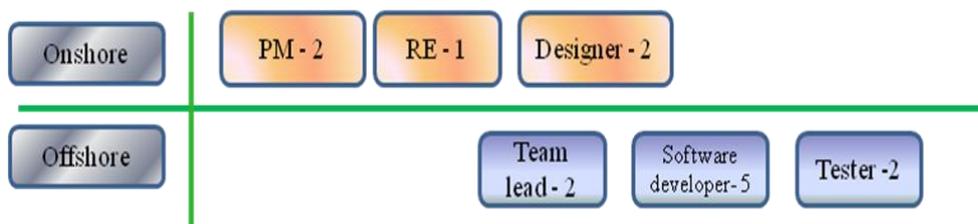
The software product suit ‘VTouch Pro’ uses the client – server architecture. The ‘VTouch Pro’ is composed of two application components. The ‘VTouch Pro Server’ is a server component. The server application broadcast an advertisement/message to one or more client machines located at customer end. The client application ‘Digital Display System’ DDS receives advertisement /message from central server and display it on digital signboards, screens and monitors installed at the customer site.

### 3.1.3 Customer / Client

The Mermaid has vast range of customers. The customers of Mermaid are the shopping stores, aviation firms, businesses firms and universities etc. The most prospective customer of Mermaid is the shopping stores in Scandinavia. Few of the clients of Mermaid are the Leaf Denmark, Apotekernes, Louis Poulsen, Seven Eleven and Carlsberg etc. For example, the leaf Denmark has 65 stores which hold more than 1200 screens in their shopping malls.

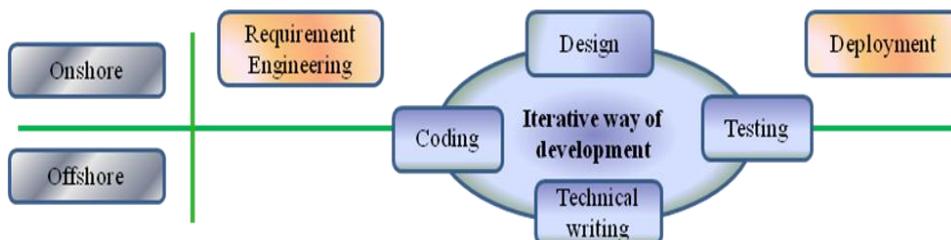
### 3.1.4 Distributed Team Classification

The Mermaid development project consists of two teams, the onshore - Danish team and the offshore - Pakistani team. The onshore team consists of the project manager, the requirement engineer and the software architecture/designer while the offshore team includes the senior software developer as the team lead, software developers and the testers. The total number and the classification of team members at the two sites are described in the figure 3.1.



**Figure 3.1. The classification of team members at the onsite and offsite location**

Similarly, the software development activities at both sites are described in the figure 3.2. The requirement engineering and the deployment activities are completely performed at the Danish site since it requires close discussion with the client and the installation of an application at the Danish business site. While the code, design and the testing activities of software development are performed iteratively between the team members of the both sites.



**Figure 3.2. The software development activities in the study project**

## 3.2 Scrum Implementation

In this section, the authors discuss the company Scrum development activities and as implemented by the studied company roles.

### 3.2.1 Roles

In the distributed project of the company, the Scrum team works at geographical separated two offices in Denmark and Pakistan. The authors focus on the development team which is composed of project manager, team leaders, software developers and testers. In this part, there is a short description about the development team, which is as follow:

- **Project Manager**

The project manager has more than 5 years of experience in the company. Project manager works at the Danish office and is close to the client. Project leader is the top authority and has vast knowledge about the client requirements. He maintains regular communication and receives feedback from the clients.

- **Team Leader**

There are two team leaders at the Danish office and they also have more than 5 years experience in the company. The team leaders are close to off-site team and are usually involved communication with them. The team leaders act as Scrum Masters in the project.

- **Software Developers**

There are five software developers in Mermaid technology. These software developers worked at Pakistan office. The developers team in Pakistan office lead by a senior software developer which has 4 years of experience in the company.

- **Tester**

Similarly, there are two software testers in the project. Software testers worked at the Pakistan office. The company follows pair programming development methodology; hence the Software testers have an important responsibility in the project. They worked as programmer cum tester in the company.

### 3.2.2 Documentation

Documentation is also paid significant attention in the project. The team documents all development processes and daily development tasks like SRS, System architectures, and test cases. In addition to regular development documents, the following reports have been implemented in the company, following Scrum practises:

- **Daily Scrum Report**

The developer teams at Pakistan office sent daily report to the team leader. It is usually informal report. The developer's tell the team leader about the today's task. The following are the usually questions.

- What have I done today?
- What are the problems in today task?

- What can I do to perform my work as efficiently as possible?
- What will I do tomorrow?

- **Weekly Report**

On every Friday, the remote team members sent a weekly progress report to the team leader. The team leader at the Danish office keeps record of the finished tasks. Finished tasks information is written down in the Burndown chart, which is maintain and updates regularly. The weekly report consists of following information:

- A brief report on what has been done during current week.
- Coming week task overview
- Weekly issue report
- Weekly risk report
- Miscellaneous

- **Sprint Review Report**

The Sprint report comprises the description about completed tasks. The sprint report depicts the current status of ongoing project. The contents of the report are the product features, finished tasks, expected risk etc. The team leader sent this report to the product owner once a sprint comes to an end. The aim of the sprint report is to notify the customer about the progress of ongoing software project. The report consisted of the following information:

- Sprint summary. A sprint report and issues rose during the sprint.
- Sprint deliverables. Information about the tasks in the current sprint.
- Sprint burn down chart.

- **Mid Sprint Report**

The team leader sends out a status report to the product owner. The report had the same content as the sprint review report. The aim of report is to notifying the product owner about the progress of the sprint.

### 3.2.3 Meetings

The communication between the team members between two offices is an important activity in the company. Although, the team members communicate and coordinate each other regularly, however the following formal and informal meetings are organized.

- **Weekly Meeting**

A weekly meeting held among the developers, testers, the team leader and the Project manager every Monday morning. The participants of this meeting are the off-site and in-site team members of the project. This meeting usually lasts for one hour and are organised over 'Skype' voice channels. The agenda of the meeting is the current week task. Tasks are defined in the meeting and are assigned to the team members.

- **Mid Sprint Meeting**

The mid sprint meeting held at the Danish site and are usually face to face communication. This meeting held between the project managers and the project client. The agenda of this meeting is to discuss the current progress of the sprint.

- **Sprint Review Meetings**

The Sprint Review meeting is also held at the Danish site. The participants of the meeting are the project manager and the product owner. It has been developed as a combination of sprint demonstration and sprint retrospective. The meeting agenda were generally focusing on what tasks had been done and not, and the reason for this.

- **Lunch Break Meeting**

The lunch break is held at both offices, in which employees get their lunch together. The ‘Combine lunch’ is a social activity to promote friendly working environment. The rule in this gathering is not to talk about work. In this gathering, the employees get close to each other.

### 3.3 Communication Support

To overcome problem of virtual team members in the distributed project, the company uses some emerging approaches. The most intuitive approach to alleviate distance is to adopt communication technologies. In this section, the authors discuss communication support provided in the distributed project. The following figure summarizes previously described activities and the usage of these technologies as enablers for communication.

**Table 3.1. Scrums Meetings and Mode of Communication**

Scrum meetings	Mode	Communication			
		Face to Face	Voice	Instant Messenger	Email
Weekly meetings	Distributed		√		
Mid Sprint meetings	Collocated	√			
Sprint Review meetings	Collocated	√	√		
Lunch Break meetings	Collocated	√			

#### 3.3.1 Voice Communication

Voice communication is an important activity in Mermaid technology. It is considered as active and efficient way to close together with remote site. To set up bridge between members of two offices, the company uses ‘Skype’ as voice communication media.

Skype is a computer-to-computer voice service, which provides free long distance internet calls. According to information from the company employees, it is easy to use and has good voice quality which helps team members to easily and clearly communicate with each other.

All team members have Skype ID and are linked to another geographical separated office through this media. The Scrum master and Scrum leaders at the Danish site communicate to development team at Pakistani office through this media. Similarly, the clients also communicate to Scrum master and development team through Skype

and at sometime through phone call. Skype is such an effective media that even the developer's team at same location uses Skype voice call in some urgent case and for early response.

Skype is considered as effective media in the company. Every Monday morning, there is formal meeting among Scrum team members over Skype. Similarly, Meetings between client and developer team also held over Skype.

In the following sub-sections, the authors contribute more detailed information about how online meetings are deploying in the company. It also discusses about how Skype's voice communication is practices at remote site. The authors categorized 'communication issues' into two group's technical problem and user related problem relating to voice communication.

### **3.3.1.1 Technical Problems**

In this section, the authors provide an overview of technical problems that were investigated in an interview held through the audio conference with the members from Pakistan.

- Digital disturbance occurred during conference meeting. Sounds like internet noise intervene during a meeting.
- Another observed incident was that surround sound from the office disturb the conference participants. The sound was either beeps or loud voices.
- During conference the participant voices gave echoes in the room. Due to this, a sentence or few words have eliminated at receiver end.
- The conference participants started talking at the same time, which create confusion.
- The main problem at one of the office was the frequent electricity breaks down, due to which the voice communication between two the sites suffered a lot.

Further the use of inadequate computer equipments at the remote site obstructed the voice communication among the team members. As it was observed in the studied project that:

- Several times when starting a meeting team members were not able to speak or even hear their remote colleague. This occurred due to some minor problems like broken headset, a headset not connected correctly to computer or wrong setting of the Skype application etc. Sometimes the cause of the problem was unknown. This problem was solved by re-calling to all participants of meeting.
- One of the participants' computers halted or restarted due to which he had to wait a little while to rejoin again the group meeting.

### **3.3.1.2 User related Problems**

In this section, the authors highlight some user related problems that arise in voice communication.

- The employees at the two offices of Mermaid Technology have different culture and language. English is used as intermediate language to communicate with other site. As English is not the mother tongue for either of the sites and also employees had poor skills in English language speaking, sometime participants miss conversation part or interpret it wrongly. This makes participants unclear over some points in discussion.

One more thing observed in these online meetings was the ‘lack of decisiveness’ which low down the utility of project discussion.

- During weekly Scrum meetings, there did not have any moderator at the remote site. When the Project manager asked question from participants of the meeting, silence establishes. In this situation, the team leader at the remote site has to answer the question.
- No agenda was set up for the meetings. Usually, the meeting agenda was prepared ad-hoc before the start of meeting. This agenda was never saved anywhere. Thus, sometimes important points/issues are missed during discussion.

### 3.3.2 Instant Messenger – IM

Instant messaging (IM) has become a significant way of communication in global software development. Instant messaging (IM) is not only used as chatting for fun among young people but is also considered as efficient communication tool for distributed projects. Given the functionality IM tools provide, such as low delay in message delivery and peer-to-peer messaging, the authors expect IM to provide substantial benefits to communication in distributed projects by lowering the barrier to communicate, and by increasing the overall awareness of presence between two sites.

In this section, the functionalities and properties IM tool provides in distributed project are focused. How the employees at Mermaid technology make use of IM functionalities and properties. The authors will also highlight the issues that arise due to its use in distributed project.

Instant messaging tools commonly provide means to make one’s presence status available for others, through online or offline identifiers. In addition to this, team members manually adjust their presence status by setting ‘Away’, ‘Unavailable’, ‘In meeting’, ‘Lunch break’ etc. The purpose for this kind of information is to make sure the current working status of team member is transparent to the remote team members.

*“There is a need for a tool like IM, where you can see the Off-site team members available” [Project manager]*

As instant messaging is a lightweight way of communication, it allows team members to communicate while simultaneously working. The multitasking ability was often used in meetings, both co-located and distributed. Instant messaging was used as a way to get answers to questions during the meeting, or to communicate issues raised during the meeting. One interviewee reported actively using IM as a side channel during the meetings.

*“IM is a lightweight way of communication. I do my testing job and communication with remote site simultaneously”. [Tester]*

One of the most important aspects of instant messaging is naturally that it is instantaneous. IM tools do not impose long delays between messages. It allows team members to ask questions and know that they are going to be answered in a timely manner. It is a more suitable tool for communicating short and simple questions in comparison to other media such as electronic mail or phone.

*“IM is useful in cases you need a quick reply to something”. [Software Developer]*

In order to notify remote side team member, instant messaging tools be used as auditive or visual alerting. IM alerting functionality is used as notifying the off-site team member.

*“IM has an alerting functionality. I use it to ping another team member and to ask some urgent or important” [Software Developer]*

Conversations over instant messaging tools generate a document, a transcript of the conversation. Usually, it is possible to save the conversations in IM tools, and the tools can usually be configured to save the conversations automatically. The main purpose for saving the logs is to keep some kind of evidence of the decisions made through IM.

*“We often have conversation with Project manager and even client as well; we save the logs from the chat. Later on, if team leader ask some technical detail, we Review and share information in the log” [Software Developer]*

### **3.3.2.1 Limitation in use of IM**

IM is successfully used by all employees either working in-site or off-site at the Mermaid Technology. The authors rarely found negative attitudes or experiences regarding the use of instant messaging. However, there are few situations where the slighter use of IM had a bit bad attitude over team working.

- In general, team members clearly identified the need for ‘presence status’ functionality in instant messaging tool. In many cases, the lack of awareness of the situation and the misuse of the ‘Availability status’ of IM lead to misunderstanding.
- It seemed to be difficult for team members to identify when the transcripts of chat should be saved and shared with others. As the nature, seriousness and content of conversation changed rapidly.
- One of the issues in multitasking with IM is when a team member is engaged in two or three distinct conversations. In such situation, the person has to keep in mind the context/background of the whole conversation with multi-people chat.

*“I usually engaged in chatting with multiple peoples at the same time. In this situation, sometimes it’s difficult for me to remember the background of conversation when communication with someone.” [Software Developer]*

- IM is used between two team members and thus all decision made have to be additionally communicated through the email to all other interested members of the team.

- The most important issue arise when communicating system requirements as text over Instant Messenger. Most of the time, the requirements were unclear. In such situation, the Project manager made requirements clear either by the help of diagram or by communicating over voice channel.

*IM is good for fast communication. It's good for a small question but not for Requirements detail. [Team leader]*

### 3.3.3 Email

E-mail is an important and unique method to communicate with remote team. It is simple to use and one written text/letter could be delivered to multiple participants in one email. People find it familiar and safe because it is familiar in much respect to written letter. To gain maximum benefits from the simple media, it is considered as efficient communication channel in Mermaid project. All the participants of the project actively communicate with each other over it. Similarly, all the formal and informal corresponding in the company took place in English language

The project management team at in-site office did formal official corresponding with remote site over email. Mostly, the management team send the official text letter to all the corresponding participants in one email. The formal corresponding between two sites is the 'official notification', 'Meeting agenda', 'Organisational rules and regulation', 'Office holidays' etc. Similarly, the management team did all official corresponding with hundreds of its clients over an email. According to one of interviewees the advantage of using email as communication media is that the formal corresponding between two persons is easily documented and is automatically saved over mail server. The document could be easily found out in the future whenever it has need.

Besides official corresponding, the informal project conversation between the members of two sites also took place over an email. The daily development task and its corresponding requirements details are send to the development team via email as an attach document. One of the advantages of using email as communication media is that not only text messages but also project SRS, design documents, organisation presentations, project movie clips, advertisement clips and flash videos could be delivered to remote site as an attached file. In Mermaid distributed project, the daily development tasks are done in the form of pair programming. One person did coding part and another did its testing. In this pair programming activity, there requires mutual collaboration and understanding between these two remote members. During this mutual activity, the remote members communicate with each other over an Instant messenger. But when requirements or traced bug description is unclear to developer, in such situation they start to communicate over an email. The developer sends requirements detail in text form to tester over email. The tester tests the required features. And send back traced bugs and there error description documented in a file as an email attachment. According to one of interviewee, "Email is an efficient way of communication. You can send a long description about the daily task and the necessary hurdles and risk in it. Rich information could be sending to another member through it".

### 3.3.3.1 Limitation in use of Email

In this section, issues are highlighted when team members communicate over email. These issues are documented after consultation with the development and management team.

- Though Email is a fast communication channel but sometimes the reply is received late from the other end.

*“Team members show laziness in replying to emails when there is an urgent need of information.” [Project manager]*

- Communication between two sites was in English language. The detail requirements are sent to development team over an email. Sometimes, developers interpret the written requirements wrongly. And after a whole day coding it, the software developer came to know that he interpreted the requirements wrongly.

*“It’s difficult to express and interpret the development task written in English language and communicated over an email.” [Software developer]*

- Every Monday morning, the project manager sends meeting agenda to participants. Similarly, the long written ‘minutes of meeting’ were sent to the meeting participants over an email. The development team rarely takes care of long written emails. They take ad-hoc overview of it and ignore irrelevant information of an email.

- When the Project manager and the Team leader has a conversation over email about the project, the developers team remained unfamiliar with the decision.

*“The Project corresponding over email remains private. When I have a communication with the Project manager over email, then I usually called a meeting at off-site office. And transfer all information to another team member.” [Team leader]*

- When writing an email, it is observed to add the necessary context/background information. Referring the background information in email makes conversation understandable. However, to do so it took much time to prepare just a single email message. Few times the problem descriptions and the requirements detail sent via email messages were unclear and were seemed as a long written document.

*“The team leader asks me about the bugs traced a month ago. I reply back to describe more information about the bugs.” [Tester]*

*“Sometimes, project management team demands unfamiliar thing in a short written email.” [Software Developer]*

## 3.4 Knowledge Exchange Support

The participatory studies focused on the knowledge exchange support in the project. In this section, there is description about collaborative work space to support knowledge exchange among geographically separated team.

### 3.4.1 Collaborative Tool

The 'Mustundant' is company owned/designed collaborative tool. The company used 'Mustundant' to share knowledge resources among development team and to know the current status of task/sprint/project. It is also used as discussion forum for development project. The use of collaborative workspace was central and is existed online. And that's why it could be easily operated/access from any location. All the stakeholder of the project (development team, clients) has access to collaborative work space and different actors have different rights in the interface. As a cooperative workspace 'Mustundant' laced in feedback and awareness of what was going on in the current project. If something is added in a folder the change was visible.

The collaborative work space 'Mustundant' was located in Denmark and therefore the response times were slow, i.e. it took a significant time to upload and save content in the workspace. This lead to another observed issue; the developer team at off-site hesitate to add content to the workspace and rather shared the files with in-site team over Skype.

The 'Mustundant' is an online collaboration interface which is used to hold the product backlog, the sprint backlog and the burndown chart. The online tool consists of discussion forum to share knowledge among team members. It also composed of the 'Bug list' feature. By utilize this feature of 'Mustundant'; the client can upload the traced bugs or error message information over it.

### **3.4.1.1 Product Backlog**

The development team used 'Mustundant' to hold their product backlog, sprint backlog and burndown chart. The Product backlog composed of product requirements, system functionality, product features and properties. The product backlog was usually developed and maintained by the Project manager along consultation with client. Common content in product backlogs were:

- ID
- Title
- Work order
- Status (New/Constructed/Requirement changed/Ready/In Sprint/Completed/Removed/Problem)
- Planned sprint
- Time estimation (Counted in days or hours)

When new items were added to the product backlog in the middle of the project they got the status 'New'. When the development 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.

### **3.4.1.2 Sprint Backlog**

Spring backlog was design and managed by Project management team at Denmark office. The common sprint backlog columns were:

- ID
- Title
- Assigned to
- Status
- Product Backlog Item
- Estimated Hours/Days

- Hours Remaining
- Comments

As Sprints build product application, the Project Manager re-estimates backlog items. The length of sprint was usually set four to five numbers of weeks. Sprint duration was estimated along consultation with developer team.

It was observed that sprint backlogs were not always populated with the proper amount of hours in the beginning of the sprint. It was because developer didn't know what should be included and only added the things they were working on. Extra hours were added to a sprint, to cover if something went wrong.

### **3.4.1.3 Burndown Chart**

The development team had sprint burndown charts at the common workspace. The burn down chart is an online chart showing remaining work in the sprint backlog.

Team member is responsible for estimating the number of hours remaining to complete all assigned tasks during a Sprint. As task be completed, new estimates are made until all work is done. These estimates are then summarized for all tasks and converted to a burndown chart which can be used to determine overall progress being made during the Sprint.

According to interviewees, the development team seldom take care of following/using the charts. The only team leader was putting together the sprint review reports where the charts were included. The team leader updated the burndown chart daily.

### **3.4.2 Team Discussion**

The development team has created online place for project discussion. It was an asynchronous thread forum. It enabled all actors in the project to post text. The forum was threaded and team members could reply to the posted items.

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.

### **3.4.3 Bug list**

Bug list composed of detail about system error messages. A bug or error message could be put up/upload online either by the development team or by the client. A bug list was a similar listing as the backlogs. It contains the following detail:

- ID
- Title
- Error message
- Assigned to
- Status (Active/Resolved/On hold/Clarification needed/Closed)
- Priority (High/Normal/Low)
- Due date
- Bug type

Usually, software testers at remote-site deal with bugs and error messages.

### 3.5 Customer Involvement in the Project

The active role of a project client in the development process is one of the significant importances in an agile manifesto. In agile methods, the project client acts as a team member [28]. Further in Scrum method, the role of a client becomes more since to act as a 'Product owner'. At one side his responsibilities includes to motivate the project team members and at the other side to design the project roadmap accord to business goal.

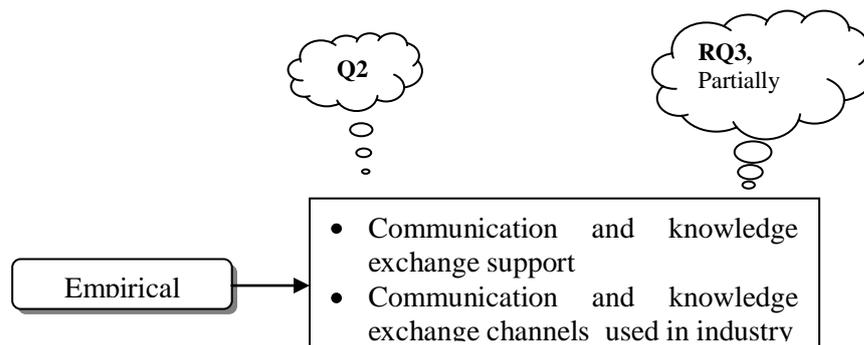
In the studied project, the authors have observed that the Danish client had much communication with the onsite team as compared to onsite working colleagues. The project client collaborates with the onsite team members in planning the product backlog, preparing estimation in backlog and to prioritise the requirements in the sprint items. Further the author observed that the Danish client had rare interaction with the offsite development team. This was the client representative had less English language skill and he feels more secure and confidence while communicating with the onsite counterpart. As one of the interviewees responded:

*"I have two or three times communication with the Danish client. He most of the time interactive with the onsite team members" [Team lead]*

## Summary

To analyse the Scrum practices in a distributed project and to insight the communication and knowledge exchange support to it, the case study is conducted in a company ‘Mermaid Technology’. The ‘Mermaid Technology’ is a Danish based company have a development centre in Pakistan. The company deals in the ‘Digital Signage System’ and have offshore development part of its business to a remote country. The Mermaid Technology has two set of teams, the management and the development team. The management team resides at Denmark and have a close collaboration with the Danish clients. However, the development team, having different culture and the language, works in the different time zone, at Pakistan office.

The development methodology in the study project is an agile base Scrum. The Scrum practices in the studied project are the Scrum roles (Scrum master, Scrum team etc), meetings (Daily meetings, Sprint meetings etc) and reports (weekly report, sprint reports etc). The Iterative development has a variable length of 4-5 weeks. The product backlog is maintained by the management personals along consultation with the clients. The management personal transfers the product information and the requirements detail to the developers through the ICT (Information and Communication Technology). The agile development demands close collaboration with client, rapid requirements change and the team spontaneous communication [2, 6, and 25]. When this way of agile approach is applied in a distributed project, it requires frequent communication and knowledge exchange among the dispersed team members and need close collaboration with the customer over distance.



**Figure 3.3: The communication exchange in agile practices**

The Information and Communication Tools (ICT) consider being the efficient way to connect the two teams together. The team member communicates each other through a voice media, email and IM messages. To share the product backlog and other project documents, the company designed its own collaborative-workspace ‘Mustundant’. The collaborative work space ‘Mustundant’ is equipped with the discussion forum, burndown chart and the bug list. In the case study, it has been observed that among all the used communication channels, the voice channels not only build-up the synchronous communication but it also help in building the trust and mutual understanding among the multi-culture peoples, which are the core in agile development. Similarly, the use of IM properties (logon/logoff, alerting) and functionalities (text, lightweight communication) help the team members to look at the availability/presence of remote partner, spontaneous messaging, audio/visual alerting and multitasking during a shared task. Despite the channels (Voice chat, IM and Email) effectiveness, it depicts few limitations in a cross information sharing. E.g. Email communication seems being an asynchronous media. It usually received an email reply on the next day whereas in agile development the spontaneous information is required in the iterative development of fixed length.

## 4 EVALUATION OF THE AGILITY LEVEL

There are many limitations that preclude the full applicability of agile practices and the described investigation uncovered some of them in the studied organisation. In order to determine whether the team at the studied company has implemented the Scrum practices, we will evaluate these by performing the Nokia test [47].

### 4.1 Nokia Test

In 2007, the Bas Vodde coached and trained the Nokia – Siemens network staff in China on Scrum and designed the Nokia test to include the Scrum practices [46, 57]. The purpose of the Nokia test was to determine either the team has implemented the Scrum practices and at what level. The Nokia test is similar to a maintenance check on a car. It looks whether a car tires have air, its tank has gas and all cylinders are firing. An owner of a car should examine these before he goes out for a drive with his Scrum team [47].

The Nokia Test consists of eight questions about how the team is adopting the Scrum. The test questionnaires are in two parts. The first part of the test questions relates to the Iterative development. The questions are:

- Iterations must be time boxed to less than 4 weeks
- Software features must be tested at the end of each iteration
- The Iteration must start before specification is complete

The next part of the test checks whether the team is implementing the Scrum practices. The questions in this part are as under:

- Do the team know about their Product owner?
- Are the product backlog is prioritized by business values?
- The product backlog has estimates created by the team?
- Are the team regularly maintains the burndown charts?
- Is there any disruption among the project team?

### 4.2 Methodology

In this Nokia test, the authors will be able to evaluate the level of application of agile development practices by the distributed team in the Mermaid technology.

In the methodology sections, we will discuss about the participatory study, the study group, the questionnaire composition and the ways of collecting data. From the experiment result, we will evaluate our finding ‘Are the distributed team is doing the agile development?’

#### 4.2.1 Survey Study

A survey is a quantitative study, and assuming the selection is made in a correct manner, a result that represents the whole population can be acquired by examining only a sample of the total population [21].

In this experiment, a survey study was performed mainly in two ways; through online - Skype interviews and through questionnaires sent by mail or posted through an online survey.

By doing a survey study, the authors are able to examine all stakeholders (the development team members, management team members, the clients) point of view over the agile development practices. The authors have the possibility to present the results in graphs and are able to relate the survey result with the Nokia test – the agility level.

#### 4.2.2 Data Collection Method

When the authors came to the decision on which method we should use to collect data for the survey study, we essentially chose between the two ways of doing the survey study; either doing survey through external online resource or sending the survey questionnaires personally. Both of these two ways of data collection method has their own benefits and drawbacks.

#### 4.2.3 Survey Participants

In order to reach our findings, the authors have conducted the Nokia test in the studied company. In this short experiment, the authors have classified the survey participants into the following two groups; the development team and the management team. The purpose of the member’s classification is to get an accurate result from the experiment. The development team participants include the team leads, the software developers and the testers personals reside at the remote site. The management team participants include the project managers, the system architecture and the requirement engineer. The total numbers of persons in participatory study were 14. The table 4.1 depicts the study groups and the members participate in the experiment.

**Table 4.1. The Persons participated in the Survey study**

<b>Development team members Offsite team</b>		<b>Management team members On-site team</b>	
Team leader/Senior developer	2	Project manager	2
Software developers	5	System designer / Flash designer	2
Testers	2	Requirement engineer	1
Subtotal	9	Subtotal	5
<b>Total = 14</b>			

#### 4.2.4 Nokia Test - Questionnaire Composition

In the Nokia test, each person on the team has to choose an appropriate selection in the eight different categories of questions [47]. The questionnaires in the test relates from the following Scrum artifacts and practices; Iteration, testing in the sprint, product backlog, product owner, agile specification, estimation, sprint burn down chart and scrum team. The survey questionnaire can be found at the appendix B.

There are eight different questions. Each question has multiple optional answers. Each selected answer has a unique score, depicting the level of the agile practices.

When composing a questionnaire, there are three main ways of asking the questions regarding type of answer: open questions, questions with fixed alternatives and questions where the answer is plotted on a scale [21, 22]. However, in this test, each answer get a unique score, ranging from 0 - 10. The lowest scale '0' resembles the 'Not agree' and the highest scale '10' depicts 'strongly agree'. Each optional answer has been marked with a unique score as according to test designer Sutherland [49, 50]. The optional answer followed by the marked score can be found at the appendix C. In this test, the individual survey result is gathered and then the whole team result is calculated and averaged to evaluate the net Nokia test score.

### 4.3 Test Result

The survey questionnaire was sent out to fourteen - 14 staff members of the company, with a deadline set of one week ahead. Out of these, nine members from the remote site and the three members from the on-site team participated in the test and hence the return rate on the questionnaire was 85%. To reach an accurate result, the authors think it is enough. Firstly, individual participant data is gathered, and then the collective team result is calculated and averaged to get the net test score. The questionnaire answers can be found at the appendix C.

From the applied Nokia test, the studied company received 4.8 out of 10 score, which means that the company follows forty eight percent – 48 % standard agile practices in their development project. Graphically it can be presented as follows (see fig. 4.1).



**Figure 4.1: The team score at the Nokia test**

The test relates to the company implemented Scrum artifacts and practices. The following eight categories (as shown in fig. 4.2) are chosen for the test. The Score of each category depicts the level of agile artifacts and practices followed by the team. E.g. the testing category shows the level five - 5 score. This means that the features of the module are tested as soon as they are completed during the iteration. The figure below describes the team agile practices and their respective score in the Nokia test.

Table 4.2. Team Scrum practices along with their Nokia test score.

Scrum Artifacts	Get Nokia Test Score	The Team Scrum Practices
Iteration	2	The iteration of sprint is variable but is always less than six weeks
Testing	5	During iteration, features of module are tested as soon as they are completed.
Agile specification	5	During the start of an iterations, the team have good agile requirements
Product owner	2	The project has the product owner but is not involved with the team.
Product backlog	7	The company possess single product backlog.
Estimates	8	The estimates in the product backlog have been produced by planning poker along consultation with the team members.
Burndown chart	5	The sprint burndown chart do not reflect state of work in progress
Team disruption	5	The company has friendly work environment. The project manger and team member have their own task. No one interrupt in others work.

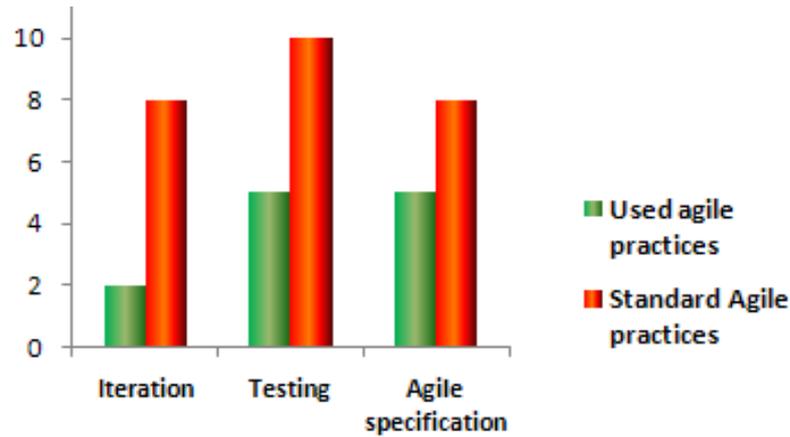
According to the Jeff Sutherland's, the designer of Nokia test, if the company implements the Scrum artifacts and implement the necessary Scrum practices, the company will get reasonable 7.0.score, which is the pretty Good Scrum level [46, 47], as shown in table 4.2.

## 4.4 Analysis and Discussion

In this section, the authors will discuss the team agile practices based on the Nokia test experiment. To analyse the results, the authors categories the team Scrum practise into two categories; the iteration in the sprint and the product backlog. This categorization is based on the Nokia test questionnaire formulation.

### • Iteration in the Sprint

In the studied project, the overall iterations duration in the sprint is variable. The average duration of sprint is less than six weeks. When the iteration phase starts, the team gets well describe agile requirements. Similarly, the team tests the features of developed item during each iteration phase. However, for efficient Scrum practices the length of the sprint should be fixed and less than four weeks. At the end of each iteration, the software should be tested and deployed at the customer end [47, 48]. The Figure 4.2 shows graphical representation of Scrum implementation at the studied company vs. Standard Scrum practices.

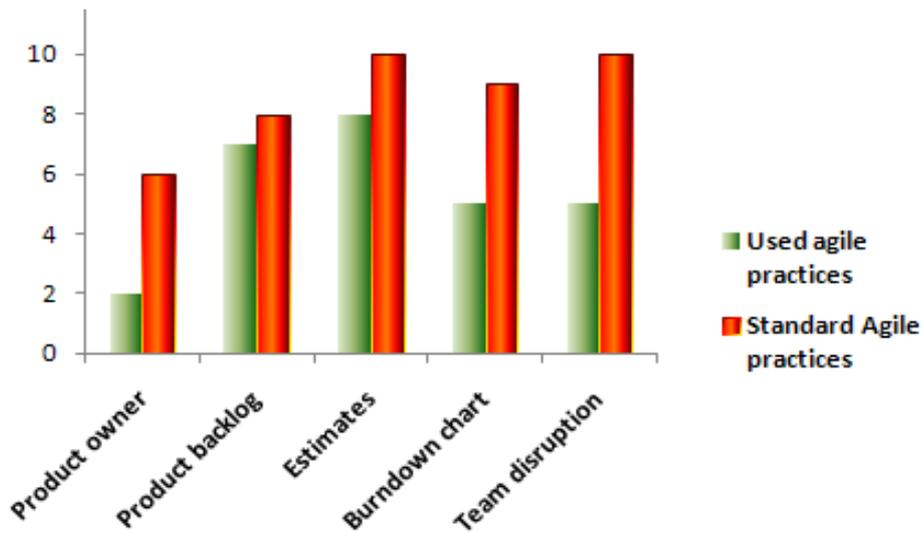


**Figure 4.2: Scrum implementation at the studied company vs. Standard Scrum practices**

- **The Product backlog**

In the studied project, the team has product owner but has limited communication with the development team. The team possesses a single product backlog. And the estimates in the product backlog are calculated by the management team members along consultation with the development team. The burndown charts was not regularly updated by the team and hence it does not reflect the state of work progress.

However, according to the Jeff Sutherland's and the Bas Vodde [47, 49], the standard Scrum methodology requires the following effective practices. The Scrum team has the product owner, who is involved in the development project and motivates the development team. The product backlog is prioritized by ROI and contains all necessary details like measurable requirements, cost per each requirement or other metrics. The product backlog has correct estimates and the possibility of errors in it should be less than ten percent 10 % [49]. The Figure 4.3 show graphical presentation of Scrum implementation at the studied company vs. Standard Scrum practices.



**Figure 4.3: Scrum implementation at the studied company vs. Standard Scrum practices**

According to Jeff Sutherland's, hyper-productive teams tend to score high on the Nokia Test. Teams that score high on the Nokia test also tend to generate much higher revenues. Jeff Sutherland uses four categories when classifying the maturity of a Scrum implementation [49]

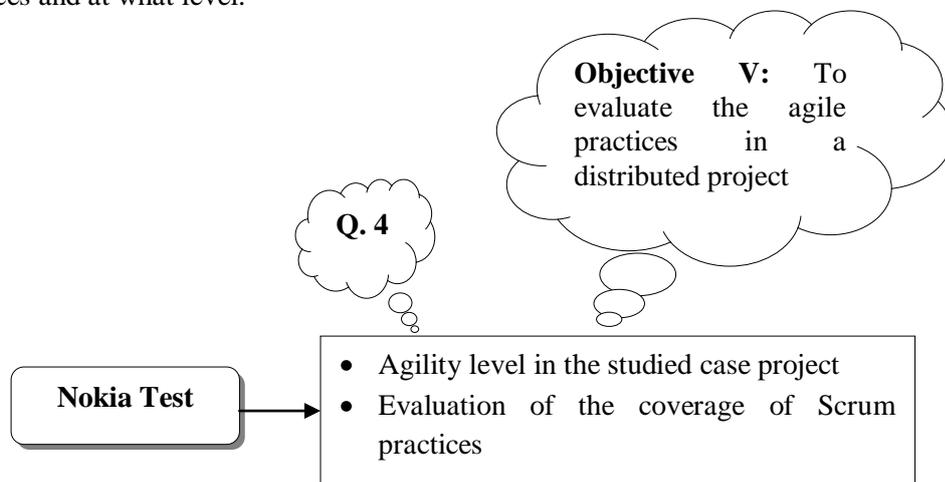
Table 4.3. Scrum implementation and Revenue earn [49]

<b>ScrumButt</b>	Revenue up 0-35%, scores 7 or less on Nokia test
<b>Pretty Good Scrum</b>	Revenue up 150% - 200%
<b>Good Scrum</b>	Revenue up 300%
<b>Great Scrum</b>	Annual Revenue up 400% !

## Summary

The agile development methodology – Scrum has standard roles, artifacts and practices. The standard practices in a project are the iterative development of fixed length, requirements elicitation; testing in iterations, task /sprint/project estimations etc. Similarly, the different Scrum roles and variables are the Scrum Master, product owner, product backlog and the Sprint backlog respectively. The Scrum methodology is designed for an in-house development, which involves simple and synchronous communication. However, in a distributed project, the agile development methodology only relies upon the tools mediated communication and follows coordination over distance. Subsequently, it's difficult to implement and fully maintain the agile practices in a distributed project.

In order to evaluate the level of application of agile practices in the studied project, the authors conducted a short experiment – the Nokia test. The Nokia test was designed by the Bas Vodde, the Scrum chief trainer at the Nokia-Siemens network. The purpose of the Nokia test was to determine whether the studied team has implemented the Scrum practices and at what level.



**Figure 4. The agility level evaluation**

A survey questionnaire was designed to get information from the team members over the Scrum practices in a project. The questionnaires were based upon Jeff Sutherland [47] recommendation. The questionnaires in the test related from the following Scrum artifacts and practices; Iteration, testing in the sprint, product backlog, product owner, agile specification, estimation, sprint burn down chart and a Scrum team. In the applied test, total thirteen members participated and hence, the result depicts eighty five percent 85% received information from the whole population.

Based on the test score gets the studied company follows forty eight percent 48% the standard Scrum practices in its distributed project. According to Jeff Sutherland [47, 48], to earn the real success in agile projects, the companies should implements the seventy percent 70% the standard agile practices in their development process.

## 5 DISCUSSION

In this chapter, the authors discuss the comparative overview of the agile methodology Scrum practices and the tools mediated communication in a distributed project.

### 5.1 Applicability of the Scrum practices in a Distributed Project

Scrum is a process skeleton, which contains sets of practices and predefined roles. The Scrum role and practices are defined and builds for a project team that works side by side at the same place [20, 30]. The processes involve an informal communication and an incremental development of a software project. The development processes are mainly designed for an in-house development project. Scrum processes are designed to add energy, focus and transparency to a project team [13, 20]. Some studies indicate that the Scrum roles and its practices cannot be implemented in a distributed project [53]. However, there are few case studies [13, 20 and 53], which depict the Scrum-roles distribution in a distance project and their successful implementation in a distributed environment.

In order to investigate the Scrum practices in a distributed project, the authors explored the offshore project and found that the Scrum practices were adopted by the distributed team. Despite the geographical distance, the project team successfully implemented the Scrum roles as Scrum master, Product owner and a Scrum team. The distributed project warmly appreciated the customer involvement in a development process as a 'Product owner'. A customer participation in a project has a core value in an agile manifesto [3]. The project adopted incremental development approach and the working prototype had to be built in a sprint of fixed duration. The project also adopted a single product backlog and the online Burndown chart was maintained to reflect the state of work progress in each Scrum-sprint. To meet the agile informal communication, the studied project promoted the regular Scrum meetings with all distributed team members. And made possible the information sharing with each other.

The GSD has challenges and issues of geographical separation, time zone differences; language and culture dissimilarities between a project team members [38]. The authors found that in the distributed environment the Scrum practices were to certain extend suffering from the global separation. E.g. Due to the physical separated team members (one of the GSD factors), the project communication and information sharing only rely upon the computer-mediated tools.

The GSD challenges delimited agile practices in the distributed project and it was also depicted through the Nokia test results. From the Nokia test conducted in the studied project, the authors found that it followed 45% of the standard Scrum practices.

Here, the authors explain how the GSD issues had effected the adoption of the agile practices.

- **Language barrier**

- The Danish client was the part of project and had much interaction with the Danish onsite-management team. But due to no spoken English language skill, the Danish business representative had limited communication and interaction with the offsite development team.

- **Distance team work**
  - No face-to-face meetings. And hence the Scrum meetings usually relied upon the computer mediated tools;
  - During the start of Sprint iteration the remote team members had unclear requirements specification. The project management personals and the customer at the Danish site firstly prioritised the requirements in the sprint and then transferred the detail requirements and its expected solution to the development team.
  - The Sprint iterations had a variable development length. And it usually took more than four weeks due to the problem understandability and in response the excessive information exchange. According to Sutherland, the Sprint iteration should be fixed length and be less than four weeks [48, 49].
  
- **Time difference**
  - The onsite and offsite development team works in different geographical region and shared different time zone. In summer time, the two teams had four hours time difference and in winter time it reduced to three hours. The working-hours time difference between two sites slow down the regular communication and information sharing between the dispersed team members [38].

From the studied project, the authors also found that although the GSD project comes with the issues and challenges, however the project team adopted such strategies and practices to meet the offshore project nature/ demand. The effective measures being taken to meet the project demand and nature are shown in the figure 5.1.

**Table 5.1. Strategies to minimize the challenges in distributed agile development**

<b>Distributed Project Challenges</b>	<b>Agile development Essentials</b>	<b>Strategies adopted to meet these challenges/requirements</b>
<ul style="list-style-type: none"> <li>• Geographical separation</li> <li>• Different languages</li> <li>• Social differences</li> <li>• Multi Culture</li> </ul>	<ul style="list-style-type: none"> <li>• Communication</li> <li>• Team coordination</li> <li>• Commitment</li> <li>• Iterative development</li> <li>• Customer involvement</li> </ul>	<ul style="list-style-type: none"> <li>• Promoted the regular and frequent information exchange between the remote team members, management personals and the customer via the computer mediated channels and through phone calls.</li> <li>• The project communication was in international business language.</li> <li>• A pair testing strategy between the distributed team members mitigate the virtual work, the teamness feelings and was considered as a effective strategy for a quality product and a short iterative sprint.</li> <li>• Adjusted +1 h in each site working-hours to gain maximum working-hours overlap.</li> <li>• The culture ambassador and to build more trust between the two sites: One of the remote team member worked more than a year at the Danish site.</li> <li>• Promoted cross site visits.</li> </ul>

From the above discussion the authors finally conclude that although the agile practices in the studied case did not increase the speed of the case studied project as it normally does in a collocated project, the implemented Scrum practices added transparency enabled rapid development and supported the customer value and satisfaction at all levels.

## 5.2 Tools-Mediated-Communication as a replacement of the Collocated Communication

Communication amongst humans is the activity of giving people information or of expressing ideas [23]. Communication plays a major role in a software development project, which composed of a legion of activities that are carried out by all stakeholders of a project [1, 23]. In a collocated software development with agile practices, the team members usually work at a same place and hence share the advantage of a physical closeness and everyday face-to-face meetings with the employees. This provides project stakeholders with easy communication and allows to share information with their working colleagues in an informal way. However, in a distributed project setting, the project team members reside far away from each other and usually communicate through the computer-mediated tools (Video conference, voice channel, Email or IM). Thus, the necessary communication and information sharing in a distributed project depend and relies upon the use of node-hood the 'communication tools'. This communication pattern in a distributed project becomes further complex due to the time zone differences and the different languages spoken by the multinational team members [23, 39].

Based on the studied case observation in the distributed project and by taking help from the literature study, the authors discuss whether the tool-mediated communication could replace the collocated communication setting in a software project. In order to investigate it the authors take the significant components of collocated communication and compare it with the studied 'tool-mediated' communication.

- **Mutual Understanding and Collaboration**

In a collocated project, the team members work at the same place and share the same cultural background and the spoken language skill. The team members easily developed the mutual understanding. This leads to a collaborative work and a friendly working environment [1, 16].

On the other hand, in the studied distributed project the author found that the project team had the virtual team work. Along with the cultural and linguistic dissimilarities, the team members at the two sites took time to know to each other. And it was a bit slow to develop mutual understanding with the other site team member. Similarly it was observed in the study case and was also discussed earlier that due to linguistic barrier the Danish customer feel better to share information and to collaborate with the onsite-Danish counterpart as compared to the offsite team members. This limited information sharing of the Danish customer with the offsite team was also due to the use of electronic communication (phone) with such development team that shares 5 work-hours overlap between the Pakistan and the Danish businesses work-hours.

- **Communication Effectiveness**

Since the information sharing among the members is based upon face-to-face in a collocated project, it is simple, concur and informal information exchange [15, 28]. Subsequently, such communication pattern is quick and effective.

On the other hand, in the studied project the authors found that the distributed team used the computer mediated tools (Voice channel, email, IM) and phone calls as a communication channel to interact with each others. The authors found that the

computer-mediated tools were effective in the distributed setting but also comes across with a few limitations.

#### – **Email**

The Email communication was found as an important and active communication pattern in the studied project. It delivered a single message to all participant members (the insite, the offsite and the customer) with no time delay. It was also found that it delivered not only the text message but also delivered the project documents and the code snippets to the receiver end.

Besides the email communication benefits in the studied project, there were also reported few limitations in their usage.

- Though an email is a fast communication channel [18] but sometimes the late replies were reported from the sender. It was due to the unavailability of the receiver and sometime due to the laziness of the member in replying back to the email message.
- Few times the problem descriptions and the requirements detail send via email messages were unclear and were seemed as a long written document. Subsequently, the message leads to ambiguity at the development site. Since, one of the team member reported:

*“It’s difficult to extract and interpret the sprint requirements and the development task written over a long email message.” [Software developer]*

**Thus, a short written email messages are unclear and its late reply slow down the communication flow.**

#### – **Voice Channel – Skype**

In the studied case, the authors found that to alleviate the geographical distance and the team virtualness in the project, the project team promoted the use of voice communication in their daily work. The team project meetings and the members’ daily discussion were carried out over the Skype – voice channel.

Besides the voice channel advantages there were also reported few limitations in their usage.

- Several times when starting the meeting team members were not able to speak or even hear their remote colleague. This occurred due to some minor problems like broken headset, a headset not connected correctly to computer or wrong setting of the Skype application etc. And sometimes the cause of the problem was unknown.
- It was also observed that during the project meetings the digital disturbance occurred over the Skype communication. The sound like the internet noise intervened during the meetings.
- Similarly, in the project meetings over voice channel, the participant voices gave echoes in the conference room. Due to this, a sentence or few words had been eliminated at the receiver end.

**Hence, a voice channels setup and an internet bandwidth speed are the challenges in the voice communication.**

- **Documents sharing**

In a collocated project, the project resources are available at a one place e.g. the project documents, the meeting reports, the test scripts and the code snippets all are kept at the same office. This lead to the easily access, the regular updating and the version control of these project documents.

To promote the regular information exchange and the documents (the product backlog, the test scripts, the code snippets etc) sharing in the project the authors found the use of collaborative work space in the studied project. The ‘Mustundant’ was the company owned/designed collaborative tool. The online portal was the repository of the project documents, meetings reports and the product backlog. It also comprised the discussion forum, the bug list and the Burndown chart in it.

Besides the Mustundant effectiveness in the distributed project, the authors found that the collaborative work space ‘Mustundant’ was located in the Denmark and therefore the response times were slow, i.e. it took a significant time to upload and save content in the workspace. This lead to another observed issue; the developer team at the offsite hesitated to add content to the workspace and rather shared the documents files with the incite team members over the IM or through an email.

The general communication needs and there competition in a collocated project vs. a distributed project could be presented as follows:

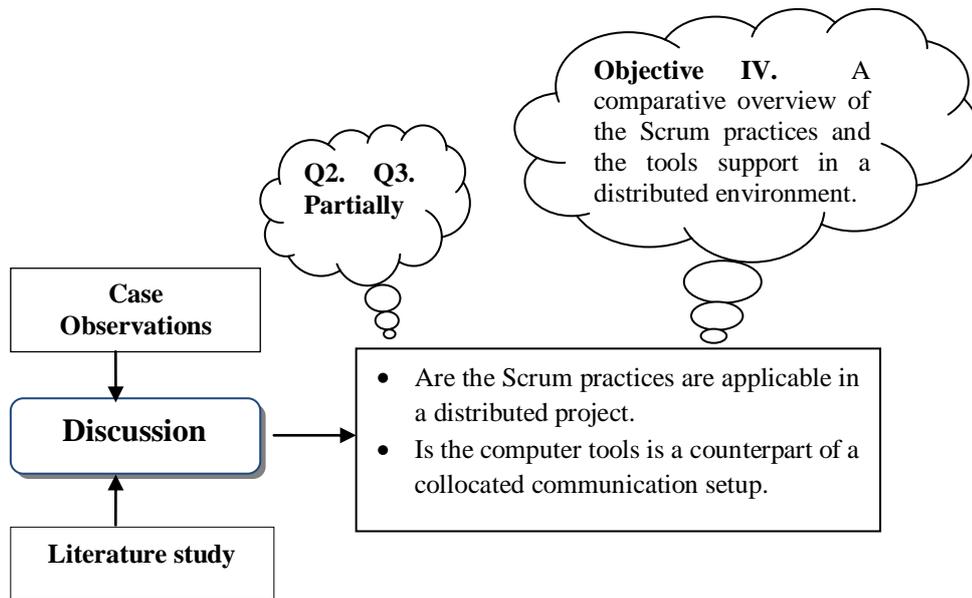
**Table 5. The communication effective in collocated and distributed project**

	Collocated project	Distributed project
<b>Mutual understanding</b>	Easily developed	Gradually developed
<b>Cooperation</b>	Due to physical closeness, the cooperation developed earlier	Cooperation over distance
<b>Communication effectiveness</b>	Due to regular face to face meetings and the background knowledge about the problem, the communication in this pattern is a quick, spontaneous and an effective.	
<b>Email communication</b>	Nil	The computer mediated communication depends upon the tools usage. This type of communication shows limitation due to human usage in the distributed project.
<b>Voice Channel</b>		
<b>Documents sharing</b>	Easily accessible and are shared at same place	Difficult to manage, shared and the version control of the documents.

From the above whole discussion, the authors finally concludes that whatever is the mode of software development (a collocated or a distributed), the communication and information sharing is a significance activity in an agile development process. In a collocated software development, the communication is easily achieved and a managed due to the nature of project environment (same place, available resources, common cultural and linguistic background), On the other hand, the information sharing in a distributed project is possible through the efficient use of ‘computer tools’ but this communication is not much effective due the tools usage issues and the project distribution limitations.

**Hence, the ‘tools mediated communication’ facilitates the information exchange setup in a distributed project but lack the close collaboration and the information synchronization as such in collocated communication setup.**

## Summary



**Figure 5: The Case Discussion**

The Agile methodology – Scrum is designed to speed up the early development of a software project. The methodology involves the informal communication and an incremental development of a software project. The development processes are mainly designed for an in-house development project. On the other hand, if this methodology is applied in a distributed project, it comes with the challenges and issues of geographical separation, the cultural and the linguistics dissimilarities between the Scrum team members.

The Scrum roles, processes and the artifacts can be implemented in the distributed project. And the regular communication among the Scrum roles is provided through the use of ‘computer mediated tools’. However, it has been seen that the global factors have adverse affect over the development processes. The customer – development team relationship and the fixed incremental Sprint are the one’s which suffered due to the slow down communication. It is a general phenomenon in a GSD, the tools mediated communication is not a counterpart to a collocated communication setup.

To summarize all, the Scrum processes did not speed up the velocity to the distributed project as such in an in-house development due to the ‘tools mediated’ communication limitations.

## 6 VALIDATION AND THE VALIDITY THREATS

In this chapter, the authors describes about the validity of the case study observations and the accuracy of the Nokia test result. Besides this, the authors mentions the anticipated threats in the case study and have described how these threats had been addressed during the case study.

### 6.1 Validation of the Result

In order to validate the finding of the research result and the suggested recommendations, the static validation has been carried out by presented the study findings to the studied project team members. The validation process is to confirm and verify the study observation and the research findings. In this section the authors will discuss how the validation process goes through.

The findings of the result were the adopted Scrum practices in the distributed environment (as described in table 6.1) and based upon them the recommendations about ‘the agile practices in distributed project’ (as described in the chapter 5.) has been setup. The applied Scrum practices in the studied project are as under:

**Table 6.1. The Project Scrum practices**

Scrum Artifacts	The Team Scrum Practices
Iteration	The iteration of sprint is variable but is always less than six weeks
Testing	During iteration, features of module are tested as soon as they are completed.
Agile specification	During the start of an iterations, the team have good agile requirements
Product owner	The project has the product owner but is not involved with the team.
Product backlog	The company possess single product backlog.
Estimates	The estimates in the product backlog have been produced by planning poker along consultation with the team members.
Burndown chart	The sprint burndown chart do not reflect state of work in progress
Team disruption	The company has friendly work environment. The project manger and team member have their own task. No one interrupt in others work.

The authors had sent these two project findings to the onsite and offsite team members, either they also agreed upon the study conclusion and findings. Moreover, the authors also got an appointment for the project presentation which has been conducted over the Skype communication.

In the project presentation, the members participated were the team lead, the software developers, the tester from the offsite team and the project manager form the onsite team. In the project presentations, the authors had presented the study project observations, the applied agile practices and based upon it the project recommendations was suggested to the team members. After presentation, the authors asked several questions from the interviewees regarding the reliability and the correctness of the results. The general questions asked were: *How much real/accurate are the agile practices observation presented in the case findings? How far is the Nokia test result when compare it with the applied Scrum practices in the project? How much important are these recommendations to the case project? Is there any*

*missing in the project recommendations? What are the few recommendations which the team members seem to be much important? Etc.* The interviewees all agreed over the study findings and are much focused upon the agile project recommendations in the distributed environment.

The interviewees commented that the important factors in establishing agile practices in distributed projects are the effective communication between the team members and the active participation of the customer in the software project.

## **6.2 Validity Threats**

The validity threats are important for any research study by identifying all key factors have dependency upon the research or can affect the accuracy of the final result [21]. According to Wohlin [58], there are four general types of validity threats in a research study namely internal, external, qualitative and construct validity threats. However, in this research study, the authors have identified the internal, external and qualitative validity threats which were relevant to this type of research study.

In this section, the authors will discuss what the expected threats in the study were and what strategies were adapted to minimize and avoids the affect of each threats upon the research study, as reflected in the following sub sections.

### **6.2.1 External Validity threats**

External validity is to what extent findings from the research study is generalized to and across population of persons [21]. According to Wohlin [58], the selection of subject and treatment is among one of the threat to external validity. The external validity threat usually occurs when wrong subjects are selected from the population which affects the final result.

- **Population Validity**

In order to examines the distributed project issues and challenges, the authors has conducted the nine – 9 interviews in total from the participatory project team members. The six of the interviews were conducted from the offsite team members and the three of them from the onsite team members. The selection of subjects in these interviews were the project manager, team leads, designer, software developer and the tester which divert different culture background and were actively involved in software development process. Hence, by interviewing the multi-national peoples, the authors become able to examine the cultural diversity and the linguistic difference affect upon the distributed project.

Similarly, in the Nokia test survey, the authors had sent the survey questionnaires of almost all team members of the project. And the total fourteen – 14 members participated in the Nokia test survey and hence it was 85% response feedback from the project team. The authors believe that it is an enough population size for getting a concrete Nokia test result.

### **6.2.2 Internal Validity Threats**

Internal validity is the relationship of causes and effects. The Internal validity is to what extent the findings of a research study precisely depict a causal relationship between

a dependent variable and its outcomes [21]. In this section, the authors will describe what the expected internal validity threats in the study were and how it was treated to draw the valid final conclusion.

One of the internal validity threats to the research study was the originality of the interviewees' conversation. The document transcript was prepared immediately after the interviewees meetings to mitigate the chance of missing any important part of conversation. Similarly, the content/transcript was send back to the person for the accuracy and the correctness of the information.

Similarly, the two of the interviews from the offsite team members were conducted in their local language. The authors do not think it as a validity threats since it increases the confidence of the interviewees and is efficient to get rich information.

### 6.2.3 Qualitative Validity Threats

A validation technique called triangulation is followed in the empirical part of study to ensure the correctness of the data and the project observations. A triangulation is a method that compares two or more sources to confirm the validity of data or the research findings [21, 58]. If there is consistency in the data provided across the various data sources, then this suggests that the data present is accurate. In this empirical study, the triangulations adopted were: the data triangulation and the methodology triangulation.

- **Data triangulation**

Data triangulation is the use of different sources of data/information on which the study results are based. In order to get information about the applied agile practices in the study project, the data/information has been collected by studying project manuals and documents, by observing the project processes and by interviewing the team members. In the project interviews, the participants interviewed were the project manager, the team lead, software developers and the testers. Similarly, information about the development practice was asked and confirmed from the others interviewees.

Similarly, the common communication issues in the distributed project have been observed and examined by interviewing the team members at both sites. E.g. If one of the interviewees complain 'no interaction of client with the offsite team' then the authors confirm the statement by verifying and confirming the statement from all others interviewees personal.

- **Methodology Triangulation**

The thesis research focus was to examine the agile practices in the distributed project. In the first part of the qualitative study, the authors did the qualitative analysis of the agile methodology – Scrum roles, artifacts and the practices. This has been done by observing the practices, studying the project manuals and by interviewing the project team members. However, in the second part of study, the authors did a Nokia test to evaluate the agility level of the project. The Nokia test findings give a clear view about the applied agile practices in the studied project. Thus by adopting the methodological triangulation the authors get real information about the studied agile project.

# 7 RECOMMENDATIONS

This chapter suggests the recommendation to implement the in-house development methodology in a distributed project.

## 7.1 Establishing Distributed Scrum Project Setup

The Scrum role and practices are defined and builds for a project team that works side by side at the same place [20, 30]. The process involves an informal communication and an incremental development in a short Sprint. On contrast, the distributed development lack information sharing and come across with the issues: the teamness feeling, the virtual work and cooperation over a distance. The authors of the thesis suggest the following recommendations for establishing an agile project setup in a distributed environment upon the studied case findings and related literature review. And the need to implement few of these recommendations has also been observed in the studied case and has been marked in the recommendation table. (Follow table 6.2).

On the basis of literature review and the studied case observation, the standard practices and a set of recommendations are listed which are helpful in establishing and maintain an agile method in a distributed project.

The recommendations are listed under the agile methodology – Scrum practices and the distributed development practices categories.

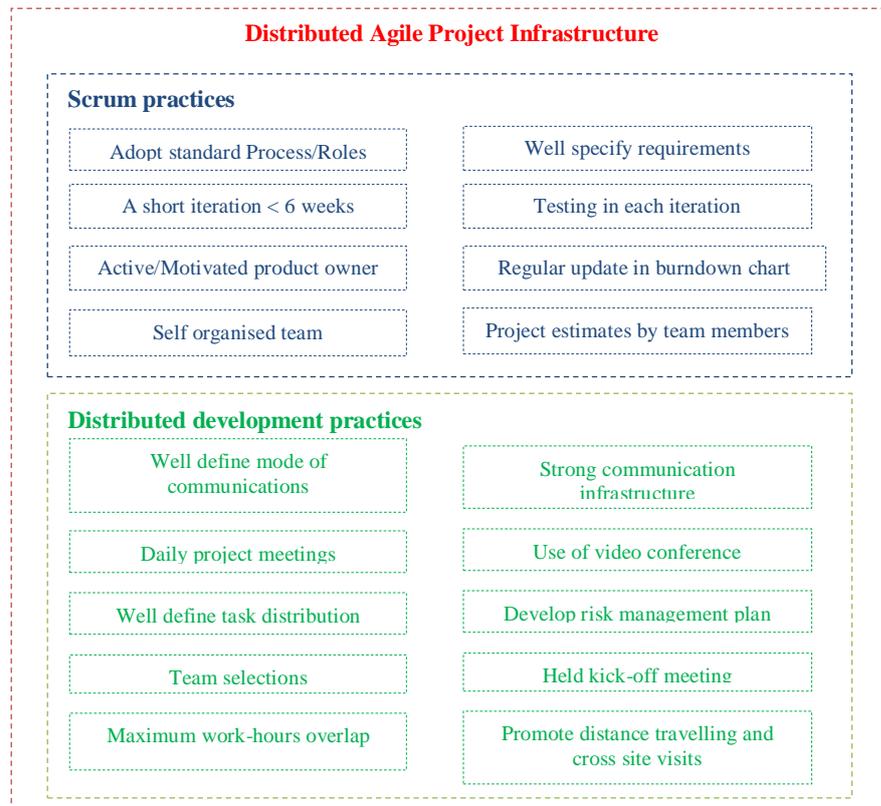


Figure 7.1. Distributed Scrum project setup

The methodology category includes the standard roles, processes, artifacts, iteration, product backlog, testing, estimation and the requirements specification set of recommendations. While the distributed development category includes the mode of

communications, distance meetings, task distribution, team selection, risk management, project milestones, distance travelling and cross site visits set of recommendations. The distributed development category includes such recommendations which mitigates to some extent the barriers in the distributed projects (like virtual work, distance communication and coordination, cultural and social dissimilarities). Subsequently, it will help in implementing properly the in-house development methodology in a distributed environment.

The researchers and industrial practitioners have suggested the following recommendations to establish the agile project setup in a distributed environment.

**Table 7.1 Distributed agile project setup**

Recommendations	Descriptions	Sources	Suggestion for studied case
<b>Agile methodology – Scrum practices</b>			
Recomd. 1	Implements all standard Scrum roles, practices and the artifacts.	[5,13,20,53,54]	√
Recomd.2	The Sprint iteration should be fixed and less than six weeks.	[47,48,53]	√
Recomd.3	Before the start of sprint iteration, the Scrum team should have users stories well tied with requirement specifications.	[47,48,53]	-
Recomd. 4	At the end of iteration the software should be tested and deployed at the customer end.	[47,48,53]	√
Recomd. 5	The product owner should have a release roadmap with dates based on the Scrum team velocity.	[47,48,53]	√
Recomd. 6	The Product backlog has estimates produced by planning poker by Team	[47,48,53]	√
Recomd. 7	The sprint burndown chart should be burns down when a task is completed.	[47,48,53]	-
Recomd. 8	The Scrum team and the project leader should be self organised.	[47,48,53]	-
<b>Distributed Software Development Practices</b>			
Recomd. 9	Strong communication infrastructure	[1,2,9,11,13,19]	√
Recomd. 10	Maximum and efficient use of computer mediated tools. E.g . Email-IM-voice channel.	[18,36]	√
Recomd. 11	Encourage the use of video conference	[18,34,35,36]	√
Recomd. 12	Promote regularly Scrum meetings.	[13,27,52,53]	-
Recomd. 13	Consider the complexity and size of project in task distribution	[13,33]	√
Recomd. 14	Consider the personal and the spoken language skill in an agile team selection.	[20,33]	√
Recomd. 15	Risk management and a backup plan	[27]	√
Recomd. 16	Arrange a kickoff meeting face-to-face or via a video channel.	[27,38,55]	√
Recomd. 17	Adjust the working hours schedule at the distributed sites to gain maximum work-hours overlap.	[34,38]	-
Recomd. 18	Plan distance travelling and a cross site visits.	[27,34,33,35,36]	√

### 7.1.1 The Agile methodology - Scrum Practices

**Recommendation 1:** “Adopt/Implement all standard Scrum roles, practices and the artefacts”.

According to Sutherland, the designer of the Scrum methodology, the Scrum project should implement the Scrum roles (Scrum master, Scrum team and the product owner), the Scrum artifacts (the product backlog, Sprint backlog and the burndown chart) and the Scrum practices (the iterative development, Sprint and the Scrum meetings). In a software development, the agile practices help-out to monitor the work progress, to meet requirements changes and makes possible early development of a software product.

**Recommendation 2:** *“The Sprint iteration should be fixed length and less than four weeks duration”.*

To enjoy the real sauce of agile project, the sprint iteration should be fixed length and be less than four weeks duration. This will made possible not only early development of a prototype software but will also increase the customer satisfaction.

**Recommendation 3:** *“Before start of sprint iteration, the Scrum team should have users stories well tied with requirement specifications.”*

The development team should have clear understanding over the problems before a start of sprint iteration. The user stories should be well defined and the requirements specification properly aligned with them in the sprint backlog. The development team at the remote site have clear understanding over the user stories and the implemented sprint requirements.

**Recommendation 4:** *“At the end of each iteration, the software should be tested and deployed at the customer end.”*

When the sprint process becomes done, the designed software should tested for remaining bugs, do functional testing and be deployed at the customer end.

**Recommendation 5:** *“The product owner should have a release schedule based on the Scrum team velocity.”*

The Scrum project should have a product owner which should actively participate in the development process and motivate the Scrum team. The product owner should properly manage the release dates of sprints based on the Scrum team velocity.

**Recommendation 6:** *“The Product backlog should have estimates produced by planning poker by the Scrum roles.”*

The Scrum project should have a single product backlog. The Product Backlog should be clearly specify and describe with ROI. The estimates in the product backlog should comprise the sprint duration, cost per sprint, and work load per member. The project manager, the developer team and the customer should prepare these estimates together in the sprint planning meeting.

**Recommendation 7:** *“The sprint burndown chart should be burns down when a task is done.”*

The team members should regularly mark the status of task. The burndown chart should be burns down when a task is complete. Subsequently, the burndown chart will help out the members and the management to monitor the overall progress of the project.

**Recommendation 8:** *“The Scrum team and the project leader should be self organised.”*

The Scrum practices are originally designed for a competent and self organized team member. The Scrum team should be skilful, independent and commits collectively to Sprint goals and the final product. Subsequently, it will release the sprints at the schedule dates with a quality product.

## 7.1.2 Distributed Software Development Practices

**Recommendation 9:** *“Plan a strong communication infrastructure between the operating sites”.*

The distributed projects come across with the issues of physically separation and the time zone difference. Subsequently, it needs proper information sharing between the Scrum roles. The communication in a distributed agile project is such importance that if it is not taken seriously in an offshore development, all practices of Scrum method like early development of an agile prototype and a short Scrum sprints will be lost. Thus, the agile project demands proper communication plan so it mitigates the loss of information in a distributed project. The ‘communication plan’ improves the quality of information between the operating sites and helps having an effective information sharing.

**Recommendation 10:** *“Make possible a maximum and an efficient usage of the computer mediated tools”.*

In a distance project, the use of computer mediated tools delimits the teamness, virtual work and the communication gap between the team members. The use of audio and video conferences in a project gives face-to-face meetings feelings. The use of Instant Messenger facilitates the presence and the current status of a remote colleague. The use of online burndown chart gives the current status of a sprint or the whole project. Thus, the efficient uses of these tools strengthen the communication bridge among the Scrum roles in a distributed environment.

**Recommendation 11:** *“Encourage the use of video conference in a distributed Scrum project”.*

The Scrum project lacks the teamness feature in a distributed project. The Scrum project requires close collaboration, informal and synchronizes communication in development process. So, the use of video conference in a project will provide synchronous information exchange and will improve the collaboration over the distance. Further, it disannuls the teamness feelings.

**Recommendation 12:** *“Plan and keep regular Scrum meetings with the remote team.”*

The daily Scrum has a significance importance in a distributed Scrum project. In a distributed project, it should be keeping short and be held via a voice or through a video channel. All team members including the product owner should be participate in it. Usually, in this meeting, following questions should be discussed, what a member had task on yesterday; what his today task and what are the risks and the hurdles in implementing the tasks. The daily Scrum depicts the current status of the sprint and explicitly builds the common understanding among the dispersed Scrum members.

**Recommendation 13:** *“Consider the complexity and the size of a project in a remote task distribution to a Scrum team”.*

In a distributed project, the development processes are to some extent relates to a complexity and a size of a project. It a bit difficult to handle a complex and a large size distributed project with an agile approaches. The complex and a large size project subsequently increases the ‘problem understandability’ overhead and requires maximum information sharing with the dispersed team members. Hence, these two variables should be considered when plan a distributed agile project infrastructure.

**Recommendation 14:** *“Consider the personal skill, the background information about the product and the spoken language skill while selecting the remote Scrum team”.*

Since the Scrum sprint comprises the short duration development and the task are completed in a fixed period of time by a Scrum team member. Hence the sprint does not sustained the lack of professional skills or the loss of information sharing due to no knowledge about the problems or due to the less spoken language skill.. In selecting remote Scrum team members, the personal skill, the background knowledge about the product and the spoken language skill should be considered when initiating a distributed agile project.

**Recommendation 15:** *“Perform the risk management in a distributed project and always prepare the backup plan”.*

In a distributed agile project, a lack of teamness, a virtual work, a breakable communication and incremental development are among the few factors that have risk in a software development process. Hence, in a distributed agile project it is important to perform risk management in collaborating with all Scrum members. And there should be a backup plan when unforeseen happen in a short sprint process.

**Recommendation 16:** *“Arrange a kickoff meeting either face-to-face or via a synchronous channel at the beginning of a distributed project”.*

The poor socialization, no face-to-face meetings and the lack of remote cultural awareness are the causes in failure of a GSD projects. One practice to address it can be arranging a handshake meeting in which team members get familiar to know each other. It future facilitates informal coordination and helps in building trust among team members.

In this kickoff meeting the team members get know to each other. Further, roles, responsibilities, development practices and the mode of communication can be discussed in this meeting

**Recommendation 17:** *“Adjust the working hours schedule at the two sites to gain maximum work-hours overlap among the distributed team members”.*

The Scrum practices require close collaboration and demand working side-by-side. In a distributed environment, a project suffers a time zone difference between the operating sites. To gain maximum work-hours overlap among the dispersed team members, the working hours schedule could be sufficiently adjusted at the two sites. By adjusting it, the team members will be able to use maximally the synchronized communication channels (audio/video) as compared to others communication media (Email or IM). Subsequently, it will improve communication effectiveness.

**Recommendation 18:** *“Always plan distance travelling and a cross site visit in advance”.*

To overcome the cultural, social differences and to build a more cooperating team, a cross site visits has significance in a distributed project. Hence, a travel expense, remote country cultural familiarity should be considered in advance in a distributed project.

## **8 EPILOGUE**

### **8.1 Conclusion**

The information exchange in the distributed agile project was studied in this thesis. The authors were mainly concerned with how the communication channels support to the agile distributed project. The thesis research was designed as qualitative study in nature. The first part of the research comprised the literature study concerning the agile development model – Scrum roles, practices and the artifacts implantation in a project. The second part of the research comprised the empirical study of GSD project by performing semi-structure interviews with the dispersed team members and by conducting the Nokia test survey to examine the agility level of the Scrum project.

The result of the research depicts that although the Scrum method is primarily designed for an in-house software development where team members work together in a same place however by the use of computer mediated tools the methodology could be implement in a distributed environment. Further, the communication and the information sharing in a GSD project has such significance that if it cannot be given importance in a distributed project, all practices of agile methodology like early software development, rapid requirements changes and the customer satisfaction would be fruitless. The authors conclude that the continuous and frequent information exchange is significant in the mutual work of the geographical separated sites.

The second part of the empirical study was the Nokia test. This test evaluated the Scrum practices and artifacts maturity level in the study project. The result of the Nokia test depicts that there are two Scrum practices which are less matured i.e. the Scrum iteration development and the product owner participation in the development process. The test result concludes that the Scrum iteration phase comprises of more than four weeks during and the project client has less interaction with the remote development team. The reason was the client usually feels confident and secure while communicating with the peoples having same language and culture similarities

On the bases of case project findings, the authors conclude that in order to gain true benefits from the agile rapid development, it is necessary to adopt all standard Scrum roles, practices and the artifacts in offshore development. Similarly, to implement the Scrum method in offshore development, it is necessary to develop the communication infrastructure and the information exchange plan between the two development sites.

### **8.2 Future work**

In this thesis study, the authors had main focused on the computer mediated tools support to the Scrum practices in a distributed project. Accordingly the future work should focus on investigation which particular communication channel should be focus or which information exchange pattern could be mostly applied for establishing and maintaining the Scrum practices in a distributed environment.

The other related area of research could be to identifying factors have an affects/relates to the Scrum practices in a distributed environment. Since the Scrum practices are designed for in-house projects that require close collaboration, regular information sharing and regular progress monitoring. Hence, the future study could be

the collaboration patterns among the Scrum roles, a Scrum project/sprint/task planning and there estimation and monitoring the whole Scrum processes/the roles work in a distributed environment.

The other area of future research could be to study the Scrum practices in a **nearshore project**. A nearshore project that locates in the same geographical region and hence shares the same time zone. In such project, one could study the nearshore Scrum project that acquire frequent cross site visits, cultural ambassadors at the customer site and the communication tools information exchange. In such project, the agility level of nearshore project could be measured and to compare results set with the offshore agile project setup.

## 9 REFERENCES

- [1] Korkala, M. Abrahamsson, P. 'Communication in Distributed Agile Development: A Case Study'. *In proceedings of a conference on Software Engineering and Advanced Applications* at Lubeck, Germany. Pp 103-110, 28 - 31 Aug, 2007.
- [2] Nisar, M.F.; Hameed, T., "Agile methods handling offshore software development issues," Multi topic Conference, 2004. *Proceedings of INMIC 2004. 8th International*, vol., no., pp. 417-422, 24-26 Dec. 2004.
- [3] Cottmeyer, M., "The Good and Bad of Agile Offshore Development," *Agile, 2008. AGILE '08. Conference*, vol., no., pp.362-367, 4-8 Aug. 2008.
- [4] Lee, W.A.; Powell, G., "Extreme selling in the early stage space," *Agile Conference, 2006*, vol., no., pp.7 pp.-382, 23-28 July 2006.
- [5] Gopal, A.; Mukhopadhyay, T.; and Krishnan, M. S., "The role of software processes and communication in offshore software development", *Commun. ACM* 45, 4 Apr. 2002.
- [6] Sauer, J., "Agile Practices in Offshore Outsourcing – An Analysis of Published Experiences", *In: Proceedings of the 29th Information Systems Research Seminar in Scandinavia, IRIS 29 - Paradigms Politics Paradoxes*, August 12-15, Helsingoer, Denmark, 2006.
- [7] Helena, H.; Eoin, O. C; Par, J. A; Brian, F., "Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance," *Global Software Engineering, 2006. ICGSE '06. International Conference on*, vol., no., pp.3-11, Oct. 2006.
- [8] Eoin, O. C; Helena H; Par J. A; Brian F, "Exploring the Assumed Benefits of Global Software Development," *Global Software Engineering, 2006. ICGSE '06. International Conference on*, vol., no., pp.159-168, Oct. 2006.
- [9] Herbsleb, J; and A. Mockus, "An Empirical Study of Speed and Communication in Globally Distributed Software Development," *IEEE Transactions on Software Engineering*, vol. 29, pp. 481-494, 2003.
- [10] Hazrati, V. "Results of Agile Adoption Survey 2008", Available on internet, <http://www.infoq.com/news/2008/05/agile-adoption-survey-2008>, last visited on May 2009.
- [11] Lucas, L. Williams, L. Damian, D. Bures, H., "Essential communication practices for Extreme Programming in a global software development team", *Information and Software Technology*, Volume 48, Issue 9, Special Issue Section: Distributed Software Development, Pages 781-794, ISSN 0950-5849, Sep 2006.
- [12] Ken, C., "Agility and Requirements", Advanced Technology Systems Corporation, Sept 10th, 2008.
- [13] Sutherland, J. Schoonheim, G. Rustenburg, E. Rijk, M. "Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams", *Agile, 2008. AGILE '08. Conference*, ISBN: 978-0-7695-3321-6, pp. 339-344, 4-8 Aug. 2008.
- [14] Rayhan, S.H.; Haque, N. "Incremental Adoption of Scrum for Successful Delivery of an IT Project in a Remote Setup", *Agile, 2008. AGILE '08. Conference*, ISBN: 978-0-7695-3321-6, pp. 351-355, 4-8 Aug. 2008.
- [15] Sauer, J., Agile Practices in Offshore Outsourcing – An Analysis of Published Experiences, *In: Proceedings of the 29th Information Systems Research Seminar in Scandinavia, IRIS 29 - Paradigms Politics Paradoxes*, August 12-15, Helsingoer, Denmark, 2006.
- [16] French, A.; Layzell, P., "A study of communication and cooperation in distributed software project teams," *Software Maintenance, 1998. Proceedings. International Conference on*, vol., no., pp.146-154, 16-20 Nov 1998.

- [17] Sauer, J., 'Agile Offshore Outsourcing - Concepts and Practices for Flexible Integration of Offshore Development Services'. In: *Proceedings of the Agile Business Conference 2006*, November 7-9, QEII Conference Centre, London, UK, 2006.
- [18] Kornstädt, A.; Sauer, J., 'Mastering Dual-Shore Development – The Tools & Materials Approach Adapted to Agile Offshoring', In *Software Engineering Approaches for Offshore and Outsourced Development, First International Conference, SEAFOOD 2007*, Zurich, Switzerland, February 5–6, 2007. Revised Papers Bd. 4716, Springer, 2007, S. 83–95, 2007.
- [19] Brien, A. J.; Alfano, C.; Magnusson E., "Improving Cross-Cultural Communication through Collaborative Technologies". Stanford University, Örebro University.
- [20] Sutherland, J.; Viktorov, A. ; Blount, J. ; and Puntikov, N., 2007. 'Distributed Scrum: Agile Project Management with Outsourced Development Teams'. In *Proceedings of the 40th Annual Hawaii international Conference on System Sciences* (January 03 - 06, 2007). HICSS. IEEE Computer Society, Washington, DC, 274a.
- [21] Creswell, W.J., "Research Design: Qualitative, Quantitative, and Mixed Methods Approaches" *Second Edition, Sage Publications*, London, UK, 2002.
- [22] Seaman. C., "Qualitative Methods in Empirical Studies of Software Engineering", *IEEE transactions on Software Engineering*, Vol 25(4), pp. 557-572, 1999.
- [23] Dybå, T.; Dingsøy, T., "Empirical Studies of Agile Software Development: A systematic review", *Inf. Softw. Technol.* 50, 9-10 (Aug. 2008), 833-859.
- [24] Aspray, w.; Mayadas, F.; Vardi. M.Y., "Globalization and offshoring of software - a report of the ACM job migration task force", *ACM, Journal*, 2006.
- [25] Norwood, J.; Carson, C.; Deese, M.; Johnson, N.J.; Reeder, F.S; Rolph, J.; Schwab. S., "OFF-SHORING: an elusive phenomenon", A Report of the Panel of the National Academy of Public Administration for the U.S. Congress and the Bureau of Economic Analysis, National Academy of Public Administration, January 2006.
- [26] Khan, S.; Niazi, M.; Ahmad, R., "A Readiness Model for Software Development Outsourcing Vendors," *Global Software Engineering*, 2008. ICGSE 2008. *IEEE International Conference on* , vol., no., pp.273-277, 17-20 Aug. 2008.
- [27] Young, C. Terashima, H. "How Did We Adapt Agile Processes to Our Distributed Development?", *Agile*, 2008. *AGILE '08. Conference*, ISBN: 978-0-7695-3321-6, pp. 304-309, 4-8 Aug. 2008.
- [28] Beck, k.; Grenning, j.; and *at el*, "Agile Manifesto", available on internet (<http://agilemanifesto.org/>) last access on May 2009.
- [29] Schwaber, K., "Scrum Development Process", *Advanced Development Methods*, 131 Middlesex Turnpike Burlington, MA 01803.
- [30] Schwaber,k.; Beedle, M., " Agile Software Development With Scrum", Prentice hall, ISBN-13: 9780130676344.
- [31] Murphy, C., "Adaptive Project Management Using Scrum", available on internet (<http://www.methodsandtools.com/archive/archive.php?id=18>), last visited on March 2009.
- [32] Deemer, p., Benefield, G., "An Introduction to Agile Project Management with Scrum", available on internet (<http://www.rallydev.com/documents/b-scrumprimer.pdf>) last visited on March 2009.
- [33] Hofner, G.; Mani, V.S., "TAPER: A generic framework for establishing an offshore development center," *Global Software Engineering*, 2007. ICGSE 2007. *Second IEEE International Conference on* , vol., no., pp.162-172, 27-30 Aug. 2007.
- [34] Therrien, E., "Overcoming the Challenges of Building a Distributed Agile Organization," *Agile*, 2008. *AGILE '08. Conference*, vol., no., pp.368-372, 4-8 Aug. 2008.
- [35] Carmel, E.; Agarwal, R., "Tactical approaches for alleviating distance in global software development," *Software, IEEE*, vol.18, no.2, pp.22-29, Mar/Apr 2001.

- [36] Cannizzo, F. Marcionetti, G. Moser, P., "Evolution of the Tools and Practices of a Large Distributed Agile Team," *Agile, 2008. AGILE '08. Conference*, ISBN: 978-0-7695-3321-6, pp. 513-518, 4-8 Aug. 2008.
- [37] Herbsleb, J.D.; Moitra, D., "Global software development," *Software, IEEE*, vol.18, no.2, pp.16-20, Mar/Apr 2001.
- [38] Moe, N. B. and Šmite, D. 2008. Understanding a lack of trust in Global Software Teams: a multiple-case study. *Softw. Process* 13, 3 (May. 2008), 217-231. DOI=<http://dx.doi.org/10.1002/spip.v13:3>.
- [39] Eoin, O. C; Helena H; Par J. A; Brian F, "Exploring the Assumed Benefits of Global Software Development," *Global Software Engineering, 2006. ICGSE '06. International Conference on*, vol., no., pp.159-168, Oct. 2006.
- [40] Ågerfalk, P.J; Fitzgerald, B.; Holmström O. H; Conchúir E.O, "Benefits of Global Software Development: The Known and Unknown". In: *Making Globally Distributed Software Development a Success Story: Proceedings of 2008 International Conference on Software Process*. 2008. p. 1-9.
- [41] Helena, H.; Eoin, O. C; Par, J. A; Brian, F., "Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance," *Global Software Engineering, 2006. ICGSE '06. International Conference on*, vol., no., pp.3-11, Oct. 2006.
- [42] Ebert, C.; De Neve, P., "Surviving global software development," *Software, IEEE*, vol.18, no.2, pp.62-69, Mar/Apr 2001.
- [43] Battin, R.D.; Crocker, R.; Kreidler, J.; Subramanian, K., "Leveraging resources in global software development," *Software, IEEE*, vol.18, no.2, pp.70-77, Mar/Apr 2001.
- [44] Maria P; Casper L, "Could Global Software Development Benefit from Agile Methods?" *Global Software Engineering, 2006. ICGSE '06. International Conference on*, vol., no., pp.109-113, Oct. 2006.
- [45] Maria, P, What Is Difficult In GSD Projects? *Software Business And Engineering Institute, Helsinki University Of Technology*.
- [46] Sutherland, J., '7 Ways to Fail with Scrum'. Published in *Crosstalk, the Journal of Defense Software Engineering*. Vol. 20, no. 4. April 2007. Available on internet, <http://old.crisp.se/scrum/seminar/20071009/slides/Jeff-Sutherland-7-Ways-To-Fail-With-Scrum.pdf>, last visited on July 2009.
- [47] Stevens, P. "Nokia Test Results: Is anybody really doing Scrum? Scrum Breakfast", available on internet, <http://www.scrum-breakfast.com/2008/06/quick-poll-results-is-any-body-really.html>, <http://jeffsutherland.com/scrum/nokiatest.pdf>, last visited on August 2009.
- [48] Sutherland, J., "Nokia Test: Where did it come from"? *Scrum Log Jeff Sutherland*, available on internet, <http://jeffsutherland.com/scrum/2008/08/nokia-test-where-did-it-come-from.html>, last visited on August 2009. and [http://www.cedur.se/nokia\\_test2.html](http://www.cedur.se/nokia_test2.html)
- [49] Sutherland, J. "Money for nothing", *Agile 2008*, Toronto 8th August 2008, available on internet, <http://jeffsutherland.com/scrum/Agile2008MoneyforNothing.pdf>, last visited on August 2009.
- [50] Yunwen, Y.; Kumiyo N.; Yasuhiro Y, "Reducing the Cost of Communication and Coordination in Distributed Software Development". *An article published in book Software Engineering Approaches for Offshore and Outsourced Development*. Pp 152-169.
- [51] Trauth, E.M.; Huang, H.; Morgan, A.J.; Quesenberry, J.L.; Yeo, B. "Investigating the existence and value of diversity in the global it workforce" Available on internet (<http://cis.ist.psu.edu/files/File/Trauth/Publications/Investigating%20the%20Existence%20and%20Value%20of%20Diversity.pdf>) last visited on July 2009.
- [52] Navarro, L.; Prinz, W.; Rodden, T., "Towards open CSCW systems," *Distributed Computing Systems, 1992. Proceedings of the Third Workshop on Future Trends of*, vol., no., pp.4-10, 14-16 Apr 1992.

- [53] Sutherland, j.; Schoonheim, G.; Kumar, N.; Pandey, V.; Vishal, S. "Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India," in *Agile 2009*, Chicago, 2009.
- [54] L. G. Hayes, "Everything you know about offshore outsourcing is wrong". Datamation magazine, Feb 28, 2003.
- [55] Cusick, J.; Alpina P., "A Practical Management and Engineering Approach to Offshore Collaboration," *Software, IEEE*, vol.23, no.5, pp.20-29, Sept.-Oct. 2006.
- [56] Cooper, C.; Duce, D.; Younas, M.; Li, w.; Sagar, M.; Blair, G., "The Open Overlays Collaborative Workspace Environment", Oxford Brookes University, Lancaster University.
- [57] Ramesh, B., Cao, L., Mohan, K., and Xu, P. 2006. Can distributed software development be agile? *Commun. ACM* 49, 10 (Oct. 2006), 41-46.
- [58] C. Wohlin; P. Runeson; et al; *Experimentation in Software Engineering an Introduction*, Kluwer Academic Publishers, London..
- [59] Uy, E.; Rosendahl, R., "Migrating from SharePoint to a Better Scrum Tool," *Agile, 2008. AGILE '08. Conference*, vol., no., pp.506-512, 4-8 Aug. 2008.
- [60] Moe, N.B.; Šmite, D., "Understanding a lack of trust in Global Software Teams: a multiple-case study", *Softw. Process* 13, 3, 217-231, May. 2008.
- [61] DeLone, W.; Espinosa, J.A.; Gwanhoo Lee; Carmel, E., "Bridging Global Boundaries for IS Project Success," *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on*, vol., no., pp. 48b-48b, 03-06 Jan. 2005.
- [62] Humphrey, W. S. 1991. Recent findings in software process maturity. In *Proceedings of the European Symposium on Software Development Environments and CASE Technology (Königswinter)*. A. Endres and H. Weber, Eds. Springer-Verlag New York, New York, NY, 258-270.
- [63] Jacobs D., "Accelerating Process Improvement using Agile Techniques", Auerbach Publication, New York, 2006.
- [64] Berczuk, S., "Back to Basics: The Role of Agile Principles in Success with a Distributed Scrum Team," *AGILE 2007*, vol., no., pp.382-388, 13-17 Aug. 2007.
- [65] Binder j., "Taking time to define communication rules can be a good start" Global project management organization, available on internet ([http://www.globalprojectmanagement.org/index.php?option=com\\_content&task=view&id=33&Itemid=44](http://www.globalprojectmanagement.org/index.php?option=com_content&task=view&id=33&Itemid=44)) last visited on Aug 2009.
- [66] Krasner, H., "The ASPIRE approach to continuous software process improvement," *Systems Integration, 1992. ICSI '92, Proceedings of the Second International Conference on*, vol., no., pp.193-201, 15-18 Jun 1992.
- [67] Upadhye, A., "Understanding Cultural Differences in Offshore Outsourcing", Advisor on offshoring and senior consultant at Alsbridge plc. Available on internet, <http://offshore-programmers.com/understanding-cultural-differences-in-offshore-outsourcing/>, last visited on Nov 2009.

## APPENDIX A: INTERVIEW QUESTIONS

The interviews conducted were semi-structured interviews. The interview questions are divided into two categories the 'Introductory Questions' and the 'Domain Specific Questions'. The 'Introductory Questions' comprises the interviewees work information, information about the company, the software development product etc. Similarly, the 'Domain Specific Questions' comprises the methodology adopted in the project, process/artifacts implemented, the information exchange between two sites and the GSD factors effects over the distributed development.

### Introductory Questions

1. At what position you have working in the company?
2. What is your job responsibility?
3. For how long have you been working in the company?
4. Are you involved in more than one project at the same time?  
If YES: How do you plan and manage it?
5. A brief history of the company.
6. What is the current product of the company?

### Domain Specific Questions

7. What is the agile development methodology you adopt in your development project?
8. What are the various processes in your development model?
9. What are the different team roles in the project?
10. What are the different artifacts in the project?
11. Explain briefly about the processes and the activities in the development methodology.
12. How you manage the short iterative development.
13. How do you manage the following agile practices in the distance project?  
The sprint, product backlog, daily meetings, project reports and the burndown chart.
14. Who are usually involved in preparing the product backlog, planning the sprint and the project estimation?
15. How the tasks are distributed among the dispersed team?
16. How regularly the dispersed team members exchange information with each other?
17. Is the communication between the insite, offsite team members and the customer are planned or it happens randomly when it's needed?
18. Which communication methods do you use mostly? Why? Which one you preferred most?
19. Being a far away, how the insite team socialized with their remote partner?
  - i. Do they meet? Is there any face-to-face meeting?
  - ii. Cross site visits? Ambassador?
20. To what extent, the members of the project are aware of each other's competence?
21. How helpful do you find your tools with information sharing with your remote colleagues. Which tools?
22. How often you use these tools? And which tool you mostly used for an effective communication?
23. Is the unnecessary communication with the remote partner delay the project task?
24. Did there any misunderstandings or conflicts happen between the team members or with the customer due to cultural/language diversities?
  - a. YES: Is there any mechanism to prevent or resolve them?
25. What are the main considerations in building a corporative and a self organized team for distributed project?
26. How do you analyze finished sprint to resolve s the sprint delay/coordination pattern/communication-related issues in future?
27. Do the standards/process/conventions cause unnecessary overhead or delays?

## APPENDIX B – THE SURVEY QUESTIONNAIRES

The survey questionnaires presented here are taken from the Jeff Sutherland – Nokia test [48, 49]. In this survey, the authors intend to investigate the studied company attitude towards Scrum ways of working.

1. **Describe your iterations**

- We do not use iterations.
- Iteration length is > 6 weeks
- Iteration length is variable but < 6 weeks
- Fixed iteration length 6 weeks
- Fixed iteration length 5 weeks
- Fixed iteration length 4 weeks or less

2. **Describe your testing**

- We do not have any dedicated QA
- At the end of an iteration software is unit tested
- At the end of an iteration software is feature tested
- Features are tested as soon as they are completed (during iteration)
- At the end of an iteration software passes acceptance testing
- At the end of an iteration software has been deployed at customer

3. **Describe your requirements at the time an iteration starts**

- We do not have any requirements
- We have a big requirement documents
- We have poor user stories
- We have good requirements
- We have good user stories
- We have agile specifications
- We have good user stories tied to agile specifications as needed

4. **What about your Product Owner?**

- We have no Product Owner
- We have a Product Owner that do not understand Scrum
- We have a Product Owner who disrupts Team during iterations
- We have a Product Owner who is not involved with Team
- We have a Product Owner with a Product Backlog
- We have a Product Owner with a release roadmap with dates based on Team velocity
- We have a Product Owner who motivates Team

5. **What about your Product Backlog?**

- We have no Product Backlog
- We have multiple Product Backlogs
- We have a single Product Backlog
- The Product Backlog is prioritized by ROI
- Product Owner has a release plan based on product backlog
- Product Owner can measure ROI based on real revenue, cost per story point, or other metrics

6. **How about your estimates?**

- The Product Backlog is not estimated
- The Product Backlog has estimates not produced by Team
- The Product Backlog has estimates not produced by planning poker
- The Product Backlog has estimates produced by planning poker by Team
- Our estimate error is < 10%.

7. **How about your burndown charts?**

- We have no burndown charts
- The sprint burndown chart is not updated by the Team
- The sprint burndown chart do not reflect state of work in progress
- The sprint burndown chart only burns down when a task is done
- The release burndown chart only burns down when a story is done.

**Does the Team know its velocity?**

- Yes
- No

**Is the Product Owner release plan based on known velocity?**

- Yes
- No

8. **Team disruption**

- There are Managers or Project Leaders that disrupt Team during iteration
- There are Managers or Project Leaders that disrupt Team during iteration
- The Product Owner disrupt Team during iteration
- Managers, Project Leaders or Team Leaders are assigning tasks
- We have both Project Leader and Scrum roles
- No one is disrupting team, we have only Scrum roles

**Thanks for taking the time to fill out this questionnaire**



## APPENDIX C – NOKIA TEST SCORE

Each answer in the test questionnaire has been marked with unique score. And the test scores are marked according to the Jeff Sutherland Nokia test guideline [48, 49].

1. **Describe your iterations**
  - We do not use iterations. : 0
  - Iteration length is > 6 weeks : 1
  - Iteration length is variable but < 6 weeks : 2
  - Fixed iteration length 6 weeks : 3
  - Fixed iteration length 5 weeks : 4
  - Fixed iteration length 4 weeks or less :10
  
2. **Describe your testing**
  - We do not have any dedicated QA : 0
  - At the end of an iteration software is unit tested : 1
  - At the end of an iteration software is feature tested : 5
  - Features are tested as soon as they are completed (during iteration) : 7
  - At the end of an iteration software passes acceptance testing : 8
  - At the end of an iteration software has been deployed at customer :10
  
3. **Describe your requirements at the time an iteration starts**
  - We do not have any requirements : 0
  - We have a big requirement documents : 1
  - We have poor user stories : 4
  - We have good requirements : 5
  - We have good user stories : 7
  - We have agile specifications : 8
  - We have good user stories tied to agile specifications as needed :10
  
4. **What about your Product Owner?**
  - We have no Product Owner : 0
  - We have a Product Owner that do not understand Scrum : 1
  - We have a Product Owner who disrupts Team during iterations : 2
  - We have a Product Owner who is not involved with Team : 2
  - We have a Product Owner with a Product Backlog : 5
  - We have a Product Owner with a release roadmap with dates based on Team velocity : 8
  - We have a Product Owner who motivates Team :10
  
5. **What about your Product Backlog?**
  - We have no Product Backlog : 0
  - We have multiple Product Backlogs : 1
  - We have a single Product Backlog : 3
  - The Product Backlog is prioritized by ROI : 5
  - Product Owner has a release plan based on product backlog : 7
  - Product Owner can measure ROI based on real revenue, cost per story point, or other metrics :10

6. **How about your estimates?**
- The Product Backlog is not estimated : 0
  - The Product Backlog has estimates not produced by Team : 1
  - The Product Backlog has estimates not produced by planning poker : 5
  - The Product Backlog has estimates produced by planning poker by Team : 8
  - Our estimate error is < 10%. :10
7. **How about your burndown chart?**
- We have no burndown charts : 0
  - The sprint burndown chart is not updated by the Team : 1
  - The sprint burndown chart do not reflect state of work in progress : 2
  - The sprint burndown chart only burns down when a task is done : 4
  - The release burndown chart only burns down when a story is done. : 5
- Does the Team know its velocity?**
- Yes : 3
  - No : 0
- Is the Product Owner release plan based on known velocity?**
- Yes : 2
  - No : 0
8. **Is there any disruption among the project team?**
- There are Managers or Project Leaders that disrupt Team during iteration : 0
  - The Product Owner disrupt Team during iteration : 1
  - Managers, Project Leaders or Team Leaders are assigning tasks : 3
  - We have both Project Leader and Scrum role : 5
  - No one is disrupting team, we have only Scrum roles : 10

## APPENDIX D – SURVEY RESULTS

The following table lists the answers gathered from the Survey participants. Total fourteen members participated in the test. The first column head presents the survey participants donated by 'Px' while the horizontal first row head shows the eight test questions. The table is filled with the test score get by answering the asked questions.

Participants	Q1.	Q2.	Q3.	Q4.	Q5.	Q6.	Q7.	Q8.
P1	2	1	7	1	7	5	4	3
P2	3	5	5	2	7	8	4	5
P3	4	5	1	5	7	8	6	10
P4	5	7	4	1	7	8	2	3
P5	2	5	5	5	5	5	5	5
P6	1	5	4	2	5	8	2	5
P7	1	7	4	1	7	10	7	3
P8	2	8	5	2	5	5	2	5
P9	3	5	4	2	5	5	5	5
P10	4	5	4	1	7	8	4	10
P11	2	1	5	2	7	5	6	5
P12	2	5	8	1	5	5	9	5
P13	1	4	5	1	7	10	6	7
P14	2	5	3	2	7	10	7	6