Title: Distributed Agile Development; Suitability, Challenges and Practices

Year: 2013

Author/s: Salman Shahriyari

Supervisor: Peter Rittgen

Abstract
Uncertainty in software development and business environment and the need to increase the speed of development have driven organizations to search for methods that are responsive to both change and speed. Providing iterative development, agile development involves customers and users through different phases of development, and delivers frequent releases of software to customer while receives the corresponding feedback. Using this approach, agile development thus aims at addressing mentioned issues of speed and uncertainty while developing only what customer needs from the beginning of the project. On the other hand, distributed software development is used in many organizations to reach global talent and global market. The problems associated with distributed software development such as lack of enough communication and team coherency, have forced project managers to combine it with agile to mitigate these social problems. This study focuses on distributed agile development, its suitability for a typical project and its challenges and deficiencies. Text analysis and interviews using qualitative methods are used in this scientific research work. From the theoretical view point, different text covering agile methodology, distributed development and combination of them were considered. This study covered two parts: first, an evaluation of agile and distributed development opportunities and problems to help determine whether or not distributed development is suitable for a project and second, considering the challenges once starting to use this method and practices required to regard them. For the empirical part, the focus was put on Volvo IT employees by having seven interviews with members who are currently active in distributed agile development. These interviews were used to compare and verify the finding of the theoretical part. The results of the study were categorized into two sections. In the first part, important elements required to verify the suitability of using this method are provided. The recommended factors for this evaluation are cost, productivity, customer, team structure, etc. In the second part, the challenges of using distributed agile development were categorized into four parts: (a) challenges of selected agile method, which the focus in this study is Scrum, (b) challenges with time-zone differences, (c) communication challenges and (d) finally team building challenges. The required practices to address these challenges were also provided.

Keywords: Scrum, Distributed Software Development (DSD), Agile Methodology, Global Software development (GSD), Communication, Time-zone Differences
Acknowledgements

I would like to thank my supervisor, Peter Rittgen for his support and advice for the thesis and all other professors at University of Borås who guided me through my studies. I would also like to express my gratitude to Steve Van Hoyweghen, my supervisor at Volvo IT, and all other employees who provided me with the opportunity to explore and get a deeper insight into the core business and projects of the company. I am also grateful to the interviewees who significantly contributed to this research project.

Finally, my heartfelt appreciation goes to my parents and my family who have always supported and encouraged me in all aspects of my life.
# Table of Contents

1. **Introduction** .................................................................................................................. 2  
   1.1 Background ..................................................................................................................... 2  
   1.2 Statement of problem ..................................................................................................... 3  
   1.3 Research questions ......................................................................................................... 4  
   1.4 Purpose and expected outcome ..................................................................................... 5  
   1.5 Target group .................................................................................................................. 5  
   1.6 Delimitations .................................................................................................................. 6  
   1.7 The authors’ own experience and background ........................................................... 6  
   1.8 Structure of the thesis .................................................................................................... 6  

2. **RESEARCH DESIGN** ...................................................................................................... 8  
   2.1 The research process ..................................................................................................... 8  
   2.2 Research perspective ..................................................................................................... 8  
       2.2.1 Qualitative or Quantitative Approach ....................................................................... 9  
   2.3 Research strategy .......................................................................................................... 9  
       2.3.1 The Roles of the theoretical and empirical Study .................................................... 10  
   2.4 Data collection procedures .......................................................................................... 10  
   2.5 Data analysis procedures .............................................................................................. 12  
   2.6 Strategies for validating findings .................................................................................. 12  
   2.7 Result presentation method ............................................................................................ 13  

3. **Theoretical Study** .......................................................................................................... 14  
   3.1 Key concepts .................................................................................................................. 14  
       3.1.1 Agile Software Development ...................................................................................... 14  
       3.1.2 Distributed Software development ........................................................................... 14  
       3.1.3 Scrum ...................................................................................................................... 14  
       3.1.4 Scrum Team ............................................................................................................ 14  
       3.1.5 Stakeholders in a distributed software development .............................................. 15  
   3.2 Subject Areas relevant for the Research ...................................................................... 15  
       3.2.1 User participation in Software Development Process .............................................. 15  
       3.2.2 Level of distribution .............................................................................................. 15  
       3.2.3 Rapid Application Development Methodology ....................................................... 16  
       3.2.4 Extreme Programming (XP) .................................................................................... 16  
   3.3 Previous Research ......................................................................................................... 16  
   3.4 Relevant Literature Resources ...................................................................................... 17  
       3.4.1 Distributed Agile Evaluation .................................................................................... 17  
           3.4.1.1 Agile Evaluation ................................................................................................. 17  
           3.4.1.2 Agile Principles ................................................................................................. 18  
           3.4.1.3 Criteria for evaluating Agile .............................................................................. 18  
           3.4.1.4 Distributed Development ............................................................................... 19  
           3.4.1.5 Distributed Development Definition ............................................................... 19  
           3.4.1.6 Different Distributing Level .............................................................................. 19  
           3.4.1.7 Offshore Distribution Benefits ...................................................................... 21  
           3.4.1.8 Offshore Distribution Challenges ................................................................... 21  
           3.4.1.9 Using Agile For distributed Projects ................................................................. 23  
       3.4.2 Main Distributed Agile Challenges ........................................................................... 24  
           3.4.2.1 Effective Distributed Scrum Practices ............................................................... 24  
           3.4.2.2 What is Scrum? .................................................................................................. 24  
           3.4.2.3 Scrum Teams and Roles ................................................................................... 25  
           3.4.2.4 Different Scrum Team Structures .................................................................... 26  
           3.4.2.5 Yahoo! Case ...................................................................................................... 27  
           3.4.2.6 Scrum Meetings ................................................................................................. 28
1 Introduction

1.1 Background

Selection of a proper methodology for a software development project ought to be based on the nature, structure and characteristics of the project which are in turn commensurate with the project size, budget, team structure, team members, project criticality, etc. This also marks an important step for a project manager, where it reflects the main concerns and objectives of the project (Pham, and Pham, 2011; Shrivastava and Date, 2010). The specificity of the structure of some organizations and projects therein necessitates employing agile distributed software development. It is a new trend in software development, which attempts to capture the benefits of both agile methodologies and distributed setting development. To have a better evaluation of this mixed method, it is necessary to first consider agile and distributed development separately before analyzing their combination.

Agile is an iterative way of software development where its main characteristics include: short release, collaborating with customers, receiving feedback from the costumers and finally being responsive to change. Some factors drive projects to follow agile methods. First cause is increasing the rate of change, which is related to both change in development tools and technologies and also change in the business environment. Agile is a method which is more suitable for highly dynamic environments and allows programmers to build system more quickly with better response to the change of development and business requirement in terms of budget, schedule, resources, team, and technology (Khan, Qurashi and Khan, 2011; Tudor and Walter, 2006; Poppendieck and Poppendieck, 2006). Another factor is that users and customers expect that their ideas are heard and considered for the product either by giving feedback or in a higher level being part of development procedure. Agile promotes engaging customers for gathering requirement and development procedures, delivering product within short cycle to users and customers and receiving feedback from them. By expanding customer services, companies are trying to increase both effectiveness and efficiencies (Sutherland, et al., 2007; Tudor and Walter, 2006; Khan, et al., 2011). On the other side, Global Software Development (GSD) has gained the attention of companies in the recent years where software is developed in a multi-site, multicultural, globally distributed environment. Shrivastava and Date (2010) believes that when companies planned to reach the global market, they could solve the issues of scarcity of resources and project failures by global software development. GSD has some benefits like accessing to global market, being close to the customer and hiring skilled local developers. It also has its own disadvantages like distance, time-zone differences and missing face-to-face communication, which causes problems for team collaboration (Miller, 2008; Lee and Yong, 2010).

While considering both agile methods and distributed development, it’s necessary to explore the consequences of mixing them for a software development project. First, however it isn’t possible to have a proper and recommended agile practice because of distance; distributed development could improve some aspects of a collocated agile
practice by injecting some features to the project like adding offshore experts to the team, accessing to the local market, understanding both features and needs of the market and reducing the cost of development (Miller, 2008; Shrivastava and Date, 2010; Lee and Yong, 2010). On the other side, using agile has some benefits for project. First, regardless of the distribution, because of quick development, recruiting less people, etc. agile is considered as a cost and time efficient method (Pham, and Pham, 2011; Shrivastava and Date, 2010). In addition, following an agile approach for a distributed system development could mitigate some of the problems of not being collocated. Incremental and iterative development helps to provide parallel changes and improvements to different systems at the same time. Also, by receiving feedback from environment, effective suggestions and solutions are considered in early phases to avoid false development (Ramesh, Cao, Mohan and XU, 2006). Finally, emphasizing agile on communication, both structured and informal communication, and the need for communication in agile because of required continuous integration could mitigate the problems of missing face-to-face communication. It’s also helpful for building trust between team members and increasing the transparency over the project (Shrivastava and Date, 2010; Tudor and Walter, 2006). This background provided an explanation about the main features of the method however using it brings some problems and challenges that need careful investigations and considerations. Further discussion about it along with the approach used in this research is provided in the next section.

1.2 Statement of problem

Although agile is generally believed to be a productive approach, the contrary to this statement can also be found in some cases. For instance, there are a few cases where agile has not been a successful method as reported in the literature (Cohn and Ford, 2003; Larman, 2004; Boehm and Turner, 2005; Chow and Cao, 2008; Koch, 2005). The main reasons for the failure of these projects have been claimed to be the challenges on transition and migrating to agile, misunderstanding and unfamiliarity with the method, additional cost of user involvement and the required training, dependency of agile method on high-level technical skilled developers, etc. (Cohn and Ford, 2003; Larman, 2004; Pham and Pham, 2011; Nerur, Mahapatra and Mangalaraj, 2005). On the other hand, nonetheless using a distributed software development offers some unique benefits such as low cost, ability to use global resources, etc. it has its own dimensions, which add the general cost, or make it risky to be used. The main problems in a distributed software project is the problem of communication, finding related experts for a specific problem, cultural differences, working in different time zones, integration of tasks, lack of trust, etc. (Shrivastava and Date, 2010; Sutherland, Viktorov, Blount, and Puntikov, 2007; Lee and young, 2010; Phalnikar, Deshpande and Joshi, 2009). The level of importance of each one of them, either related to agile or a distributed development method, is related to the case. As a result, evaluating the current situation of an organization and also resources available for a specific project, based on the aforementioned challenges and obstacles, is required in order to choose distributed agile method over other methods.

Once a decision is made to use distributed agile development, some main challenges are to be regarded as ignoring them directly impacts the progress and success level of the project. These fundamental challenges are associated with the distance and the
problem of communication, unfamiliarity with concept of agile methods and the method which is used for the project, team cohesion, trust, etc. The practices to tackle and solve these challenges have been discussed (Ramesh, et al., 2006; Lee and Yong, 2010; Sutherland, et al., 2007; Paasivaara, Durasiewicz and Lassenius, 2008). These suggested practices, both from literature and practical resources, could be used as a base for a distributed agile project. While employing these practices, it’s important to consider that principles of a method don’t change over time but practices are not consistent and by passing time, using different tools and technologies, structure of the team, available resources, etc. are possible to be changed. As a result, having enough knowledge about principles, challenges and project specifications is required to have an efficient distributed agile practice while by understanding the principles, it’s possible to adapt practices from other projects and cases to a specific project (Shrivastava and Date, 2010; Poppendieck and Poppendieck, 2006).

The primary objective of this research is to provide/outline the criteria in order to choose this method over other methods and then to promote required practices to overcome its main challenges. Volvo IT is considered for more investigation about this method. This company has been using distributed agile development for some years but according to my supervisor in Volvo, there are problems for some projects using this method as some managers in Volvo are evaluating the ways to move back to traditional and collocated methods. The first interviewee, who has experience working with this method from the start of it in Volvo, also believes that the success rates of these projects, based on their criteria for meeting the schedule, staying within budget, and delivering on requirements, aren’t high. As a result, they are looking for ways to improve the practices. This situation gives opportunity to explore the main reasons for it. On the other hand, there is not too much available research that a researcher, by focusing on an organization, has gone through analyzing a company’s distributed agile projects to understand the challenges and also effectiveness of applied practices. Another feature is that this report focuses on the experience of both offshore and onshore members who are directly involved in the projects to understand which challenges they are more encountering with and whether the practices that are used for them are suitable and efficient. Another struggle is to understand in which level, the characteristics of considered projects are similar to other applied projects in Volvo and other companies. By understanding the situations, principles and positive and negative points of practices suggested in literature and applied in Volvo, this dissertation research suggests ways to improve the current practices of distributed agile projects.

1.3 Research questions

To have a focus on the aim of this study and illuminate the research area, the following research question is suggested:
What are the important factors for setting up and maintaining a distributed agile team?

To deeply exploring this research question, it’s divided into these two sub-questions:
1. What issues need to be evaluated before starting to use distributed agile software development method?
2. What are the main challenges while using distributed agile method and which practices are needed for coping with these challenges and problems?
The first question aims to check the problems and suitability of using distributed agile to avoid upcoming problems and the next question tries to understand the problems and challenges after starting to follow this approach and suggests some solutions for the challenges of using this approach. The second question covers four different challenges: (a) challenges of selected agile method, which the focus in this study is Scrum, (b) challenges with time-zone differences, (c) communication challenges and (d) finally team building challenges.

1.4 Purpose and expected outcome

By following two steps, which are analyzing literatures by reviewing the related articles and then investigating into the agile problems and practices of Volvo IT, this report is trying to reach two purposes: first, contributing to the theoretical knowledge by analyzing the practical finding and comparing it with what is available in literatures and second considering the problems in practice and suggesting some solutions to the discovered problems in Volvo. Answering the mentioned research question is helpful to reach this purpose.

1.5 Target group

By considering Volvo IT Company as the case of this study and by regarding distributed agile development approach, this dissertation project targets employees in this company who are related to software development projects. The main target groups in this company are project managers and Scrum Masters; those who are currently managing a project and those who are candidates to take responsibility for mentioned or similar roles in the company. It offers a guideline for choosing an appropriate approach for their projects. The report also gives useful information to lower-rank employees like developers and testers to understand how an agile distributed project works. In addition, by generalizing this information and considering the fact that Volvo IT is classified as a large company, this project aims to give information to companies who are using this method or decided to distribute their project using agile method, especially western companies targeting countries like India, China, Brazil, etc.

As another group targeted in this report, researchers within the field of Informatics specifically researchers who are eager to gain more knowledge about agile methodologies and distributing software development could use the results of this work. It provides a basement for them to understand the challenges and necessary practices of distributing agile development. Finally, this report gives knowledge to students who are interested to agile methodologies and distributing projects as two ways which will be used more and more for software development.
1.6 Delimitations

The area of distributed agile is too broad so it’s necessary to bring some delimitations to the research area. However this research tries to bring some solutions for using tools to solve the problem of communication, the technical aspects and problems are ignored in this research while the social and human interaction problems are in the main attention. Because of the broadness of software development methods used in Volvo, especially waterfall and co-located practices, this research is concentrating on the distributed agile method and is not trying to explore the benefits and problems of other approaches.

1.7 The authors’ own experience and background

The author obtained the bachelor degree from Shiraz Azad University in Iran in the field of Software Engineering in 2008. In 2010, he started his Masters of Informatics studies in the University of Boras, as he was interested to gain more knowledge about the IT field and its contribution to the business. While studying in the university, he tried to put his focus on agile method, as he believed it would be the method, which will be used more and more in organizations. Considering it, he published his paper, “Co-Design of RAD and ETHICS methodologies; A Combination of Information System Development Methods” in “2011 International Conference on Software and Computing Technology (ICSCT 2011) “ in Singapore. After that, he started his efforts to do his Masters thesis by receiving this title from Volvo IT: “Agile methods in distributed settings; How to cope with time, cultural and geographical differences”. His background in studying software engineering and continuing it in Informatics field helps him to have a broader view about this thesis.

1.8 Structure of the thesis

This part presents a tentative structure of the thesis in words and in a model, which is provided in figure 1.
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>This chapter starts with a short description of the concepts, both related to agile methods and distributed development and is followed by stating the problem, which will be covered in this report. After that, the research questions are discussed continued by purpose and expected outcome. At the end of this chapter, the target group, delimitations and author’s experience and background related to this research are reviewed.</td>
</tr>
<tr>
<td>2. Research Design</td>
<td>In this chapter, the dominant method and strategy used for this research is discussed. Also, data collection and data analysis procedures are explained. The final parts of this section include thesis strategies for validating data and result presentation method.</td>
</tr>
<tr>
<td>3. Theoretical Study</td>
<td>This chapter describes the concepts used in this thesis while considers the analysis of literature and previous works related to the area of study.</td>
</tr>
<tr>
<td>4. Empirical Study</td>
<td>This chapter, which aims to verify the finding of the theoretical section, focuses on the structure of projects in Volvo and provides a summary of interviews.</td>
</tr>
<tr>
<td>5. Analysis and Results</td>
<td>The aim of this chapter is to analyse the results captured in theoretical and empirical studies by using the method described in chapter 2 of this study while considering the research questions.</td>
</tr>
<tr>
<td>6. Discussion</td>
<td>This chapter finalizes the thesis by providing the conclusions of the result and bringing some suggestions to the problems according to the analysis. This chapter continues with method and result evaluation and ideas for continued research.</td>
</tr>
</tbody>
</table>

Figure 1: Structure of the thesis
2 RESEARCH DESIGN

2.1 The research process

However I received the project title as a part of Volvo IT strategy to investigate more into the problems of developing software using distributed agile, I had my own power to formulate research questions and choose issues I believed is important to focus. I performed and started it by interviewing with two Scrum Masters, who are working in Volvo IT, and receiving an understanding about distributed agile practices and its problems while performing a literature review to understand what was done before. After that, I began data collection phase performing a comprehensive literature review. It helped me to form my interview structure for interviews with employees who are involving in Volvo IT agile distributed teams. The details for the analysis part and validating the finding of this project are described in the next sections.

2.2 Research perspective

Hermeneutics and positivism are two major scientific research perspectives and two epistemological clash, which are used in research methodology, as described by Wright (1971). Bryman and Bell (2011) used the term interpretivism as a perspective, which comes in contrast with the positivism and were used in some other scientific texts. In this section, the differences between them and the dominant perspective in this study are explained.

Positivism is an epistemological position and believes that the acceptable knowledge is just the one, which can be confirmed by the senses. It emphasizes utilizing the same approach used for natural sciences for studying the social knowledge. As a result, positivism believes on conducting science being value free and objective while existing a general knowledge for everyone (Bryman and Bell, 2011). It emphasizes on using existing variables and a systematic and statistical approach for showing the truth and is more related to quantitative methods (Cassell and Symon, 1995).

Hermeneutics view focuses on subjective view of the text, language and artifacts while promotes a constructive view. Subjective perspective, unlike objective, believes that it’s impossible to have a common view of our world because everyone has his own understanding of a phenomenon, which comes from his/her own previous knowledge about it. This view is a part of research procedure and affects the research procedure and results (Silverman, 1993). The impact can be seen in a text where the reader might have a different understanding from the author’s view, which it doesn’t mean that any of them is wrong. Bryman and bell (2011) explains that interpretivism, a term used as an alternative for the positivism view, distinguishes between people and the objects of the natural sciences and requires capturing a subjective understanding of the social science. Hermeneutics view is more the base for qualitative approach (Gummesson, 1991; Gadamer, 1979).

In this project, I aim to produce knowledge using interpretation of the texts and interviews. Also, for understanding the problems of following distributed agile, I need
to interact with people related to Volvo IT projects in a way that their subjective view is listened and detailed responds are received. Using above description, the dominant approach in my research is based on hermeneutic view.

2.2.1 Qualitative or Quantitative Approach

While discussing about the differences between quantitative and qualitative research strategies, the first regarded difference is that the first one employs measurement as a tool while the latter doesn’t but there are some other main differences which are related to their philosophical perspectives, related to what discussed in the previous section. Quantitative approach uses a deductive approach, which means that it is more suitable for testing the theories. As discussed before for positivism, quantitative approach emphasizes on applying natural science model on social sciences. In contrast, qualitative method is mainly used for generating theories known as an inductive approach while it has an interpretivism epistemological orientation by focusing on individuals interpreting their own world. Quantitative method has an objectivism ontological orientation, considering social reality as “an external, objective reality” but qualitative method regards society as “a constantly shifting emergent property of individual’s creation” (Bickman and Rog, 1998; Bryman and Bell, 2011).

The qualitative methods give more opportunity to go into details of human behaviors to understand the reasons of them. It offers more flexibility to the researcher for interacting with participants and for receiving more detailed responds from them by using methods like interviews, using open-ended questions. It increases the chances to find out the hidden problems within a system or an organization (Cassell and Symon, 1994; Gummesson, 1991; Bryman and Bell, 2011).

I’m following a qualitative method for this research since I need to go into details of distributed agile practices and the act of humans related to them. I also needed to interact with participants in a way that gives them enough capability to discuss about the issues related to their project.

2.3 Research strategy

By focusing on the purpose of a study and the nature of the problem that a researcher is trying to solve, the dominant strategy for a research is selected.

An explorative study is a study where a researcher is trying to explore more into an interest or a problem of a research area, which hasn’t been evaluated before and is relatively new. The researcher is performing it to better understand the problem or to check the feasibility for more research (Babbie, 1995). An evolutionary study is a research which follows and is based on the works which have done before to “form a chain of results in order to generate a composite results” (Andersen, 1994).

The character of my study is more evolutionary. Distributed agile has been using for years in organizations and however some problems of it haven’t been explored, it’s now an option for of developing software. In this research, I use the works of previous
researchers to better understand the problems of distributed agile development and to suggest some practices for utilizing it by focusing on Volvo IT projects.

2.3.1 The Roles of the theoretical and empirical Study

The data from theoretical and empirical sources are used to reach the results of this research. First, theoretical analysis was performed to capture the results of previous researchers in this area. This study was helpful to receive a view about issues, which have done and explored before, not to duplicate them, and use these accepted examples of scientific works as a base for the research. Also, it had an influential impact on designing the interview format and questions and helped me to have an overall view about the problems and the practices, which are required for having a challenging discussion with interviewees, exploring more about the issue.

The empirical study, which was mainly done by using interviews, was helpful to understand the current situation and works in Volvo IT as an enterprise. The problems of following distributed agile development and current and dominant practices and tools of the organization were explored in interviews. While I was in the empirical phase, I didn’t stop reviewing literature which helped me to have further analysis related to what I captured in empirical phase and to improve my interviews. Finally, I used the data of both studies for the analysis and discussion phases.

2.4 Data collection procedures

For collecting the qualitative data needed for this project, I used literature review and analysis as well as interviews. Literature review was the first step to collect theoretical data followed by semi-structured interviews with Volvo IT agile distributed team members to cover the empirical part of this thesis.

Literature Study
Literature study is the first step for most of researches, which provides a basis for elaborating the research questions, and helps the researchers to construct the research design. By using it, the researcher captures what has been done in the area of interest not to replicate it. It also gives information to understand issues, which are more important and have more spaces for doing research or things that has been missed. Also, it’s helpful to realize which method and strategy is more appropriate for the research.

I have tried to continuously review the literature, which helped me to repetitively compare what I received from the literature with my empirical study. I regarded two kinds of documents for this goal; found and researched documents. Found documents, which are the primary data, offer information about the procedures of following distributed agile in addition to the practical problems of it. These documents also show how distributed agile is practiced and how people are connected using tools. Researched documents are approved example of practices and consist of analysis of previous works and experiences and show the impact of them.
**Literature sampling**

Since too many textual sources are available within this area, sampling the literature gives a direction to review more related texts to the goal of the research. First, I tried to check the validity and reliability of the database of the text, not to use the ones with low credibility. On the other hand, I tried to cover all aspects of the research by searching for agile projects and distributed ones separately while focusing on the combination of them in distributed agile projects. For covering the impact of distribution, I focused on India, Poland and Brazil as three countries that are the main focus of Volvo IT distributed projects as well as this research. Finally, Some of the literatures that I reviewed were based on the suggestions I received from people who I interviewed with them.

**Interview**

Interview is a way of collecting data by having conversation with someone who is related to the topic of the research. There are some ways for constructing an interview. One is structured interview, which insists on asking same pre-prepared questions from all of the respondents. Here, all respondents receive same interview stimulus and the interviewer read out the questions as they are printed and scheduled. The type of questions is usually closed, closed ended, pre-coded, or fixed choice, which makes it more appropriate for quantitative method. The problem with it is that it’s not possible to capture detailed information from the interviewee (David and Sutton, 2005; Bryman and Bell, 2011). This kind of interview doesn’t match the aim of this study, as I required detailed comprehensive data so semi-constructed interview is used. Preparation is important before any interview so I tried to get an overall view about the roles and responsibilities of interviewees and prepared the questions related to his/her responsibility. It’s also important to be knowledgeable about the area of concept that you’re interviewing about, so I tried to first understand the concepts then I constructed my interviews. Because I followed semi-structured interviews, I gave myself the right to add some questions while interviewing and tried to have a challenging conversation with the other party. The interviews were recorded on a recorder and were converted to the text later.

**Interview Sampling**

There are two main ways for choosing the respondents for an interview: probability sampling and non-probability sampling. In probability sampling, all of the individuals in the population have enough chance to be selected but in non-probability sampling, respondents are selected base on some rules like easing of access or being rich in knowledge. Snowball sampling is a non-probability sampling approach, which certain people are selected based on their knowledge about the problem area (Trochim, 2009; Bryman and Bell, 2011). I used snowball sampling method for selecting my interviewees where my supervisor in Volvo IT connected me to some employees especially Scrum Masters working in this area and after that, by the help of them, I was connected to some other Scrum Masters and developers who had enough knowledge about distributed agile projects. The samples were selected both form onsite location, Sweden, and offsite locations like India and Poland to understand the problems in different locations.
2.5 Data analysis procedures

Analyzing qualitative data is a difficult task, which some considerations for it are needed. Trochim (2010) believes that since it’s a hard task, the key is to consider the previous works of qualitative research carefully. The required tasks are defining the research problems, developing a sampling plan, and developing the structure. By following this approach, analyzing data is much easier. The final goal of the analysis is to create a reportable and deeper meaning from the collected data. Having enough knowledge about the research is important for the analysis part.

Saunders, Lewis and Thornhill (2009) suggested three steps for qualitative analysis including reducing and codifying the data, classifying them in some categories, and finally analyzing the concepts. Bryman and Bell (2011) indicates that coding is an important step, which is the start point for most qualitative data analysis. I considered below factors, which are suggested by Bryman and Bell (2011):

1) Starting to do the coding soon to sharpen the understanding of the data and help with theoretical sampling, also “alleviating the feeling of being swapped” by the data.
2) A primary reading of the documents without any interpretation and then reading again to mark important parts and choosing the keyword for starting the coding.
3) Reviewing the code and after that regarding “more general theoretical ideas in relation to codes and data.”

In this research, the theoretical and empirical data from the text and interviews were analyzed.

The theoretical part of it consists summarizing and analyzing the text and information related to ‘distributed agile projects’ from different sources like databases, journals, websites, books, etc. These analysis had two goals; first, trying to understand the concepts related to my research area to capture the recent interests and shortages and base on that developing the research questions, and second, as my main struggle, answering the research questions by analyzing the different sources for increasing the validity.

The empirical analysis tries to create a more understandable report from the interviews. After each Interview, I converted the conversations to text and made a transcript of it. Then, the text was coded and classified into categories based on what I aimed in the research questions and then analyzing the different responds that I received. For coding, also, the responsibility and the focus of the interviewee were considered. I analyzed the interviews based on what I found in the theoretical research to have a comparison of them. Answering the research question was also important in analysis.

2.6 Strategies for validating findings

Validity is related to the integrity of the findings and the conclusions of a research. As Bryman and bell (2011) explained, some researchers believe that same criteria, which are used for evaluating quantitative research, should be considered for the qualitative research (Mason, 1996; LeCompte and Goetz, 1982; kirk and Miller, 1986) while
some others indicate that the qualitative research has a complete different nature and approach so an alternative criteria is needed (Lincoln and Guba, 1985; Guba and Lincoln, 1994).

Larsson (1994, cited in Lind, 2005) divided criteria for validating the research results into three sections: First is ‘validity criteria’ which concerns about heuristic value, consistency and the value of the data captured in empirical study. The next criterion is ‘result qualities’, which cares about meaning, structure of the report and theoretical findings. The final criterion is ‘qualities in the text as a whole’ and for it, text needs to be validated with issues like ethical value and perspective consciousness. On the other hand, Lincoln and Guba (1985) and Guba and Lincoln (1994) believe that an alternative criterion is needed for evaluating qualitative researches and used the term trustworthiness to explain the evaluation criteria for it. Trustworthiness is divided into four categories: Credibility, Transferability, Dependability, and Conformability. Credibility concerns about receiving confirmation of the content and result of the research from the members of the social world you’re investigating. My research was evaluated by my supervisor in Volvo IT and also by some other Scrum Masters in the company for conforming that I used a good practice and had enough knowledge about that social world. Transferability entails generalizing the results across social settings however it’s a problem in qualitative approach because of tendency to study small group. To reach the dependability concerns, an alternative for reliability used in quantitative approach, I used interviews with specified members which are repeatable and reliable. Finally, conformability, or objectivity, of the research is assured by not allowing my personal values impacts the phases of the research (Bryman and Bell, 2011; Lincoln and Guba, 1985; Guba and Lincoln, 1984).

I also used the checklists for appraising quality in qualitative research explained by Spencer, Ritchie, Lewis and Dillon (2003), which is available in Bryman and Bell (2011; page 400). The issues of authenticity, mentioned above, are parts of this checklist. To defend the research design criteria of the checklist, I selected qualitative method as a method suitable for this research since I needed detailed information about the people, practices, and procedures of agile distributed project. Also, the respondents were tried to be the knowledgeable members of Volvo IT projects while considering the diversity of viewpoints and responsibilities. Choosing Scrum Masters and developers from different onshore and offshore locations helped to reach this purpose. I tried to be knowledgeable about the field to capture rich and depth information from interviews.

### 2.7 Result presentation method

The results of this research are presented by using detailed analyses of data captured in the theoretical and empirical study. It also consists answering research questions. The results are presented using textual format, which is appropriate for discussing the facts and different ideas captured in different phases of a qualitative study.
3 Theoretical Study

3.1 Key concepts

3.1.1 Agile Software Development

Agile software development is a group of software development methods, which emphasizes on iterative development as a way to respond to the change. Unlike waterfall methods, customer has a key role in development process while using this approach. Customer interacts with the agile team and gives feedback about the work, while as an agile principle, teams are required to satisfy customer by early, iterative and continues delivery of software. Teams in agile should to be small, cross-functional and self-organized. Scrum and Rapid Application Development are two examples of agile methods (Pham and Pham, 2011; Shrivastava and Date, 2010).

3.1.2 Distributed Software Development

Distributed Software Development is a term used when part or all of software development cycles happen abroad, outside the origin country of the company. Companies follow this method for variety of reasons like reducing the cost, accessing to local market or accessing to skilled people of other countries. Some problems like problems in communication, collaboration, and team consistency come up while using this method mainly because of the distance, time-zone and cultural differences of teams who are working in different locations (Lee and young, 2010; Phalnikar, et al., 2009).

3.1.3 Scrum

Scrum is a trendy agile software development, which assumes that developing software involves working simultaneously on requirement, analysis, design, coding, and testing and after that delivering the entire system. The iteration in Scrum is called Sprint and after each Sprint, the team is supposed to have a release to the customer. Scrum has planned some meeting like Daily meeting, Sprint Planning meeting, Retrospective, etc. to provide a mechanism for team members to communicate more, share knowledge and investigate and recognize problems early on (Sutherland, et al., 2007).

3.1.4 Scrum Team

Scrum Team (agile team) is a phrase used to describe the development team working with Scrum (agile) method. The composition of members of these teams are varying in different Scrum teams and projects but they usually include developers, testers, Scrum Master, Architect, etc. They are locating in onshore, offshore or in both offshore and onshore locations depending on the structure of the team (Pham and Pham, 2011; Miller, 2008).
3.1.5 Stakeholders in a distributed software development

There are different people and teams involved in a distributed agile development project to make it possible that the project works smoothly. One is development team, which is known as Scrum team or agile team. Business team or business people are other stakeholders involved in a project. Project manager, development manager, Product Owner, etc. are members working with business side. Finally, there is customer and the level of interaction between customer and business and development team is important for the quality of the product and satisfaction of the customer (Pham and Pham, 2011; Miller, 2008).

3.2 Subject Areas relevant for the Research

3.2.1 User participation in Software Development Process

Participating users and customers in the development process has advantages of assuring that users’ needs are met. It’s also useful to avoid future user resistance, to obtain user commitment and in general to achieve user satisfaction and have a workable system (Tudor and Walter, 2006; Cavaye, 1995). Some issues need to be considered for participating users. First issue discusses about the users who are necessary to participate in the development process. Since, there are different layers of users, it’s required to check if it’s enough that high-level management customers be involved in the development or from direct end-users, there should be one or some representatives. Second issue considers the level of participating; whether customer and users should be involved in the whole process of software development, from requirement till release, or it’s enough that they participate in the requirement gathering and evolving process. Another loose way is that users give their feedback after releasing the software (Avison, 2003; Adman and Warren, 2000).

User involvement could be a costly process considering both monetary expenses and the expenses related to user absence from their regular job. On the other side, there are some discussions that user participation will decrease the final product development cost by avoiding developing wrong and unnecessary tasks (Tudhope, 2001; Beynon-Davies, Mackay and Slack, 2001; Avison, 2003).

3.2.2 Level of distribution

There are different ways of distributing the work to the offshore team. One is partial distribution that part or the whole development process happens in the offshore location but business team is located in the onsite location near the customer. This situation works better for customer, as they are able to work in requirement and have a better collaboration with the business team. At this structure, it’s important that the development team also has enough business knowledge and has constant communication with the business team and customer. Another option is complete distribution that business team is close to the development team and works remotely with customer. There is a problem that customers don’t receive enough about the
product which results to produce unnecessary or wrong product and decrease customer satisfaction. For both of these methods, sending a bundle of requirement and expecting to deliver the software after a while without working closely with the development team results to a low-quality software (Phalnikar, et al., 2009; Shrivastava and Date 2010).

3.2.3 Rapid Application Development Methodology

Rapid Application Development (RAD) is an agile methodology, which was initiated in response to the need for developing the system rapidly. Another important factor, which was regarded by RAD developers, is responding to the change because companies weren’t satisfied on investing too much on things that give them less organizational value and will be useless after a while (Tudor and Walter, 2006). There are two kinds of uncertainties, which are explained by Beynon-Davies, et al. (2000): business uncertainty and development uncertainty. Business uncertainty is caused by new trends in business environment like virtual organization, globalization and offering customizable products. Development uncertainty discusses about increasing the rate of change related to information system development. The reusability and maintainability and the quality of the software are another important issues for organizations. Incremental development in RAD offers these features by avoiding developing wrong software and releasing better quality software for the next iterations (Agarwal, Prasad, Tanniru and Lynch, 2000; Howard, 2002).

3.2.4 Extreme Programming (XP)

Extreme Programming (XP) is an agile software development method, which like other agile methods focuses on short iterations, extensive oral communication between members and being responsive to change requested by customer. Unlike Scrum and RAD, it focuses on programming side of software development like pair programming (which requires that members work together and have comprehensive code reviews), simplicity and clarity in code and unit testing of all code (Beck, 2000).

3.3 Previous Research

I covered too many of available resources and previous works related to distributed agile practices but some of them, which were more in my consideration for the literature review, are discussed in this section.

Shrivastava and Date (2010) provides a comprehensive review about distributed agile projects. This paper first tries to capture the challenges and opportunities of both agile method and DSD. After that, the challenges of using distributed agile development are discussed. It also consists of the ways DSD and agile could work together to solve the problems of each other. Finally, the best practices and tools to overcome the problems of this method are suggested. Ramesh, et al. (2006) in the paper ‘Can distributed software development be agile’ follows the same procedure by comparing different and opposite features of distributed and agile practices like fixed quality
requirement of DSD and evolving quality requirement of agile to find out the possibility of mixing and finding a balance between them.

Lee and Yong (2010) discusses about the case study of My Yahoo! Zorro1 project. For the project, they first used heavylight methodologies but after receiving insufficient results from it, they switched to lightweight methodologies, particularly Scrum. They explained that reason for shifting to agile was slowness of heavylight methods and the need for interacting with customers. The study “focuses on how the global product team and international teams work together to internationalize and localize the 18 international version of Charmeleon using agile practices” and discusses about different roles in the project and how they are distributed in different location. It also brings the challenges of communication, control and trust, and then discusses the useful practices to consider them. Sutherland, et al. (2007) also goes in detail about the results captured by two agile companies, SirsiDynic and Starsoft for distributed Scrum practices to understand the issues, problems and practices. It also considers three different Scrum team structures and compares the advantages and disadvantages of them.

Finally, Paasivaara, et al. (2008) discusses about the challenges of combining GSD and agile methods in larger projects. The paper considers “a case study on agile practices in a 40-person development organization distributed between Norway and Malaysia”. It also covers suggestions for practices and tools to have a better working environment.

3.4 Relevant Literature Resources

3.4.1 Distributed Agile Evaluation

In this section, after providing short definitions about agile methodology and distributed development; the benefits and challenges of each of them and the combination of them are discussed to have an evaluation for choosing it for a software development project.

3.4.1.1 Agile Evaluation

3.4.1.1.1 Agile Definition

Software development methodologies are divided into two main different categories: heavyweight and lightweight methodologies. Heavyweight methodologies consist of traditional approaches like Waterfall Model and Spiral Model and focus on comprehensive documentation, inclusive planning and extroverted design. On the other side, lightweight methodologies like Scrum and XP, which are known as agile methodologies, allow programmers to build system more quickly with better response to the change of business requirement in terms of budget, schedule, resources, team, and technology. These methods focus on iterative development of short life cycles while involving the customers for gathering the requirement and participating them as a team member for development phases (Khan, et al., 2011).
Agile process model, which is a specification of lightweight, emphasizes on faster and nimbler software development processes. Agile doesn’t try to follow a pre-specified rigid plan but focuses on good design for improving the quality of software. It’s more responsive to change and believes that good software architecture is one which is easy to be changed since software should expect user interaction jobs, the unexpected jobs, which will be figured out in the future, and the last minute jobs (Tudor and Walter, 2006; Poppendieck and Poppendieck, 2006). Iterative development has a key role in agile methods to make it possible that requirements and solutions evolve between team members (Shrivastava and Date, 2010). Customer should interact with the agile team and work with them throughout the development preferably face-to-face. The team also tries to satisfy the customers by “early and continuous delivery of valuable software” (Khan, et al., 2011).

3.4.1.1.2 Agile Principles

Understanding agile principles is helpful to receive knowledge about the main concepts of it. Seven agile principles are defined by Pham and Pham (2011): (1) customer satisfaction through early and continuous delivery of software, (2) accepting change in every level, which is important for customer satisfaction, (3) delivering working software frequently, (4) interaction of business people and developers when working together for the project, (5) face-to-face conversation as the most effective method of conveying information, (6) simplicity, and finally (7) self-organizing teams as the way for developing the project. Pham and Pham (2011), by using ‘agile manifesto’ published in 2008, listed agile values as “Individuals and interactions over processes and tools, working software over comprehensive documentation, constant collaboration over contract negotiation and responding to change over following a plan”.

In general, the principles don’t change over time but practices could change by passing time, moving from one environment to another and using different applications and tools. The most efficient approach for applying practices is by understanding the principle and by learning it; it’s possible to adapt practices from other organizations and previous experiences to current cases. Projects should be careful about it since there is a long history of applying a practice without understanding the principle and seeing a mediocre result (Shrivastava and Date, 2010; Poppendieck and Poppendieck, 2006).

3.4.1.1.3 Criteria for evaluating Agile

The selection of one model over other models is driven by project size, budget, team size, criticality of the project and some other factors. Regarding agile, it has some benefits for the system which include using less time for delivering the software, saving time and cost because of quick development, productivity with fewer people in short time, easier measurement of the progress and having smaller teams (Pham, and Pham, 2011; Shrivastava and Date, 2010). On the other side, project managers should be cautious about some factors when they want to choose agile for their projects. First, when the customer has some business goals but she is not certain about the requirements and asks to participate in the process of gathering requirements, agile
methods are suitable since the requirements are flexible to be changed at any stage. As a result, agile practices are best applicable to projects with uncertain requirements and high level of change since while agile guarantees appropriate level of security; the risk and time to market are decreased. Also, if working with different stakeholders like customers and users is important for a project, the answer is agile (Sutherland, et al., 2007; Tudor and Walter, 2006; Khan, et al., 2011). Agile could be cost effective by avoiding producing unnecessary software and focusing on what customer needs from the start of the project however some practices of agile like user involvement are costly. Also, for some methods like Scrum, the cost of training the method will increase the overall cost of development. Finally, the success of agile method depends on high-level technical skilled developers. If one of the team members leaves the group, the project will be affected hugely and the present of an uncommitted team member has a negative effect and may cause project to fail. As a result, if all of the developers are expert and the project is small enough, agile method is appropriate for it but for team with unskilled developers, a waterfall or spiral process model is more suitable (Cockburn, 2002; Pham and Pham, 2011; Khan, et al., 2011).

3.4.1.2 Distributed Development

3.4.1.2.1 Distributed Development Definition

Two factors are important for defining distribution concept: control structure and geographical location. The control structure has two dimensions: first, outsourcing which means that a company buys the software from another company, and second insourcing which means that a company provides the services through some internal projects. In addition, geographical location dimensions could be onshore, which means that all of company’s development procedure happens in the origin country or could be offshore; when a part of the development is abroad (Shrivastava and Date, 2010).

Offshore distribution, the focus of this project, is also called Global Software Development (GSD) or Distributed Software Development (DSD). This approach has its own characteristics like physical distance between teams, time-zone differences and cultural differences of teams’ members, etc. While companies are trying to reach the global market from local market, some problems like faults in projects and scarcity of resourcing have come up. Regarding the software development, one appropriate solution is using DSD teams where the software is developed in “a multi-site, multicultural, globally distributed environment” (Shrivastava and Date, 2010).

3.4.1.2.2 Different Distributing Level

There are two different structures for managing software development distributed teams. First one is partial offloading. In this structure design team is in the onshore site and the requirements are sent to the offshore team to be built. The aim is to keep business analysis and architecture teams onshore near customers and assign the lower-level tasks responsibilities like programming and testing to the offshore team. The expectation is that the offshore team sends the final work with application codes fully tested. Expensive inter-team communication is needed because of this kind of
distribution and it’s preferred that most of the communication be in written format (Phalnikar, et al. 2009). Figure 2 offers a graphical view for this structure.

Figure 2: Partial Offshoring Structure (Phalnikar, et al. 2009)

Another structure is called complete offshoring. In this model, just the customer is onshore and he/she sends required features to the offshore team and the offshore team responds with tested working code. In this model, the inter-team communication is decreased and intra-team communication is improved. Because of these features, this method is more useful for a long-term agile development. The most important communication, which is between the customer and the Business Analyst and Architect, is not face-to-face and it’s a disadvantage of this model. Misunderstood and misinterpreted requirements are one of the main reasons of failing this kind of offshore projects (Phalnikar, et al. 2009). Figure 3 shows how members are distributed in complete offshoring structure.
3.4.1.2.3 Offshore Distribution Benefits

The benefits and the main factors that force teams for following offshore distribution and using a distributed approach are listed below:

Accessing Global Markets: By expanding their market, businesses need to get expertise in those new markets. Companies could use the feature of proximity to the local market that gives them opportunities to use the knowledge of customers and local resources (Miller, 2008; Shrivastava and Date, 2010).

Using Global talents: Companies try to use the knowledge of high talented experts by hiring high quality employees (Miller, 2008).

Reducing Costs: Companies try to reduce the cost by outsourcing to regions with cheaper development cost, which sometimes results to 25% cost saving in compare to a domestic provider. On the other hand, it’s necessary to consider other costs like the cost of reducing the productivity or the cost of traveling between sites (Miller, 2008).

Increasing Productivity: Using resources in different time zones gives opportunity to response “time to market” pressure (Shrivastava and Date, 2010). 24/7 work cycles is a benefit for maximizing productivity for some projects (Lee and Yong, 2010).

3.4.1.2.4 Offshore Distribution Challenges

Some Challenges come up while using GSD approaches, explained by Shrivastava and Date (2010), Sutherland, et al. (2007), Lee and young (2010) and Phalnikar, et al. (2009):
Cultural issues: Cultural differences are related to differences in national, ethic, professional, organizational, team cultures, etc. of members working in one team, which cause conflicting behaviors and processes (Shrivastava and Date, 2010). For example, people from East of Asia are sometimes reluctant to discuss about negative issues and prefer to pass on positive information (Lee and Yong, 2010). I discuss more about the solutions for it in section 3.4.2.4.

Inadequate communication: Communication between distributed teams decreases while software development requires communication between team members. This issue affects the productivity while time-zone differences aggravate this problem (Shrivastava and Date, 2010). It increases the probability of building wrong functionality as miscommunication and misunderstanding might happen over requirements (Phalnikar, et al., 2009). Sections 3.4.2.2 and 3.4.2.3 discuss about this issue.

Knowledge management: There are problems for sharing and managing knowledge in GSD. It’s required that team members’ ideas are shared and GSD must facilitate it by using e-mail, online discussion and keeping them for later access (Shrivastava and Date, 2010). Knowledge management practices are related to sections 3.4.2.1, 3.4.2.3 and 3.4.2.4.

Strategic issues and decisions: dividing the work and decision responsibilities are related to this issue. There are also some difficulties to leverage available resources (Shrivastava and Date, 2010). This issue is regarded in section 3.4.2.4.

Project and process management issues: “High organizational complexity, scheduling, task assignment and cost estimation” issues become more important because of distributed environment characteristics and it’s the responsibility of managers to overcome it (Shrivastava and Date, 2010). It’s also difficult to synchronize work across locations (Sutherland, et al., 2007). The visibility over the project is decreased and it’s hard for project managers to capture an overall view about the project status. The latter problem is seen in collocated teams but in a distributed project, the problem is worse (Phalnikar, et al. 2009). Sections 3.4.2.1 and 3.4.2.3 discuss about these issues.

Risk Management: Software defects happen because of added complexity, communication problems and lack of group awareness (Shrivastava and Date, 2010). A research by the German Institute for Economic Research shows that outsourcing services lead to poor corporate performance while outsourcing production results in better quality work. It’s also more risky to outsource large software development projects (Sutherland, et al., 2007). Appropriate practices are required to manage the risk, which is discussed in section 3.4.2.

Increasing Cost: While often companies regard GSD as a cost-efficient approach, as discussed in section 3.4.1.2.3, overall cost sometimes increases because of hidden cost of outsourcing so understanding these costs are required. First, when some software development teams and life cycles are working simultaneously, maintaining these versions for the same product adds the overall cost. Also, the cost of transitioning to a new location should be considered and compared with the savings
from low labor costs. Other costs are cost of knowledge transferring, traveling, team building, etc. (Sutherland, et al., 2007).

**Integration and configuration management:** The integration team is responsible for assembling different tasks and queries, which are coming from different locations, required to build an integrated and cohesive product. When the communication bandwidth communication is low, the problem of integration is increased (Phalnikar, et al. 2009). This technical issue is related to the technical method and the degree and level of collaboration within team.

**Degree of documentation:** While in agile methods, documentation is considered as a less important feature; it’s required in DSD to have documents with more details because of insufficient communication (Shrivastava and Date, 2010). I cover the ways to have a balance of them in sections 3.4.2.1 and 3.4.2.3.

### 3.4.1.2.5 Using Agile For distributed Projects

While in theory, all team members for agile methods are in the same room; in practice, the situation is different. In a survey in 2008 conducted by VersionOne, 57% of respondents answered that their teams were distributed and 41% replied that they plan to combine GSD with agile. As discussed, distributed development brings some challenges. One solution to overcome distributed development problems is using agile practices. Incremental and iterative development helps to provide parallel changes and improvements to different systems at the same time. Also, by receiving feedback from environment, effective suggestions and solutions are considered in early phases and later after on. In addition, the change is applied to the small part of the system instead of change to the whole system. Finally, by allowing frequent changes, the faults are prevented early on to make it possible to direct the project to the desired goal (Shrivastava and Date, 2010; Ramesh, et al., 2006).

While in DSD, the visibility of project status is not enough, agile methods make it possible to see the problems in the early stages of the project by using short continuous iterations. Also, communication between team members, emphasized in agile, helps participants to see the short-term goals. Communication increases because of required continuous integration for the projects. Moreover, constant communication and delivery of software focusing on high quality work in agile help for building trust between people from different cultures involved in a DSD project (Shrivastava and Date, 2010). On the other side, involving customers brings some benefits to the project. The requirement is better understood and it’s prioritized based on efficiency and productivity. Finally, the transparency of the project increased not just for customer but for all partners involved in the project by delivering the work in short iterations and frequent builds (Tudor and Walter, 2006; Avison, 2003).

What discussed above is the general description about the ways that agile method could help distributed development be more effective. In the next section of this theoretical study, I explain the practices while discussing more about the problems and challenges.
3.4.2 Main Distributed Agile Challenges

In this section, the main distributed agile challenges are categorized into four parts: how to have a better distributed Scrum practice, how to overcome time-zone difference problems, how to solve communication problems and finally, how to build one cohesive and virtual team.

3.4.2.1 Effective Distributed Scrum Practices

3.4.2.1.1 What is Scrum?

Scrum is a trendy agile method, which tries to add “energy, focus, clarity, and transparency” to software developing teams while the goal is to reach 5-10 times more productivity over industry averages, which has been experienced for many collocated teams, according to Phalnikar, et al. (2009). Scrum is a team-based “all-at-once” model. An “all-at-once” development model assumes that developing software involves working simultaneously on requirement, analysis, design, coding, and testing and after that delivering the entire system (Sutherland, et al., 2007).

Scrum began at Easel Corporation 1993. Key factors for introducing Scrum are listed below:

1. “Uncertainty is inherent and inevitable in software development processes and products
2. The requirement for a new software system will not be completely known until the users have used it.
3. Impossibility to completely specify an interactive system.
4. Ambiguous and changing requirements, combined with the evolving tools and technologies make implementation strategies unpredictable.” (Sutherland, et al., 2007)

Scrum has some advantages since it’s a systematic risk reduction mechanism and a leaner software development life cycle. It’s a more adaptive project management process and is based on people’s motivation and pride (Pham and Pham, 2011). Scrum brings some benefits for improving communication because it offers a very structured way of communication and encourage that all members talk in meetings. In addition to the meetings, it’s suggested that members have more communication and collaboration while developing the software. These practices are helpful to build a cohesive team (Paasivaara, et al., 2008). Other benefits are increasing the speed of development, working closely with customers and users while receiving feedback from them and developing a performance-based work culture (Sutherland, et al., 2007). Iterations are used in all development teams and in the framework team. The benefits of it are increasing transparency between sites, frequent monitoring opportunities and revealing problems early on (Paasivaara, et al., 2008).

Iterations in Scrum are limited into Sprints. The duration of a Sprint are normally one to four weeks and usually no item is added or deleted from Sprint Backlog during a Sprint (Pham and Pham, 2011). There are some benefits and problems with longer (3-4 weeks) Sprints. It allows teams to produce more functionality at each Sprint while main Scrum meetings, Sprint Planning, Review and Retrospective meetings, happen
less in compare to shorter Sprints. The problems are that it’s possible to produce twice “wrong” functionality than a two-weak Sprint and there is less time for inspect-and-adopt cycles of the practices. A solution is that the project starts with shorter Sprints to focus on and increase the ability for delivering potentially shippable products (Deemer, Benefield, Larman and Vodde, 2010).

3.4.2.1.2 Scrum Teams and Roles

Scrum team is suggested to be a cross-functional team composed of a Scrum Master, a Product Owner and the development team, known as “the team”, with all the skills needed to build the software product like gathering, designing, editing and testing. The procedure of development starts when Product Owner receives input from different customers, or users who represent them, to make a list of requirement. This list is known as user stories, the wished-for functionality. It’s used to create a Product Backlog, a list of prioritized requirements, which consists everything needed for the system like business features and sometimes technical issues, however Product Backlog is mostly considered only for business requirements (Pries-Heje and Pries-Heje, 2011; Pham and Pham, 2011). The business team and development teams work on it and transform it to the tasks for the development teams to develop. Duties and roles of these mentioned teams and their main members are described below:

Development team
Development team consists of Scrum Master, developers, writers, and testers who collectively own the responsibility to deliver the project (Miller, 2008). The development team is “self-manage and self-organize and is responsible to estimate Backlog items or user stories on their own”. This team is “empowered to turn Product Backlog items or user stories into task to work on”. Finally, they need to keep track of project status (Pham and Pham, 2011).

Scrum Master: Scrum Master enables and guides the team to follow the agile practices and acts as the guardian of its processes. It’s important since some members find it hard to follow agile way of working and try to abandon these practices. Scrum master is required to remind the value of agile methods and suggest the best practices fitted to a project. Providing documents, as a baseline for using agile is helpful while the flexibility to adopt practices to the case should be assigned to the team by considering the suggestions and results of pervious works. For some projects, Scrum Master isn’t a full time role so the responsibility goes to one of the senior developers on the team (Miller, 2008). Organizing Sprint Retrospective and keeping track of project progress are other responsibilities of Scrum Master (Pham and Pham, 2011).

Coach: Having one person who is committed to the coach role and to keep team following agile methods is necessary. While there is a tendency between team members to ignore agile practices, team coach should guide them to follow the methods. In some projects, this responsibility goes to Scrum Master or temporary consultants (Shrivastava and Date, 2010).

Business Team
This team has responsibility for business-related issues and consists of Project Manager, Product Owner and sometimes Architects. Different stakeholders might
work with business team depending on the project. Product Owner has a critical role in the team, described below.

**Product Owner:** Product Owner, who is responsible for business values, represents the customer to help the team understand the customer requirements (Miller, 2008; Scharff, 2011; Pham and Pham, 2011). He collects requirements for Product Backlog and by collaborating with business stakeholders, tries to define product Backlog, which is mostly business requirement. Product Owner also works with Scrum Master and the team for releasing and sprint planning and guides the team toward achieving them. He also should be available for giving feedback to the team. The Product Owner receives feedback from customers and in broader view from community about the last iteration for updating and prioritizing the product Backlog and then working with the team to pick stories from it (Pham and Pham, 2011; Miller, 2008).

When there is a Product Owner in the main location, companies sometimes add the role of “proxy Product Owner” in the distributed locations to provide guidance to the team when the actual Product Owner isn’t available. Phalnikar, et al. (2009) suggests that the proxy should work with both technical and business team and have knowledge about both areas. There are some problems related to it. First, the proxy doesn’t have equal knowledge as Product Owner about the project and sometimes gives answers, which might be different from what Product Owner believes. Also, when the proxy acts as intermediate, it slows down the project and misunderstanding might happen while conveying the information. For Xebia ProRail PUB Case, they added the project manager role as a proxy Product Owner. Since the client hadn’t had previous agile experiences, he tried to “bring the organization into agile way of working” and discuss with the client about scope, progress and deadlines while the Scrum Masters focused on “Sprints, the process and the quality” (Sutherland, Schoonheim, Rustenburg and Rijk, 2008). This proxy was more acting as a coach for the team and was supporting some roles of Scrum Master.

### 3.4.2.1.3 Different Scrum Team Structures

There are three kinds of distributing Scrum teams categorized by Sutherland, et al. (2007):

**Isolated Scrum:** Teams are isolated across multiple locations and offshore teams are not cross-functional and may not use the Scrum process. This method will cause some problems in communication and teams relationships (Lee and Yong, 2010). It’s similar with the modular practice, which divides jobs into different modules and assigns each of them to one team. Isolated Scrum doesn’t work well with “novel products or projects with uncertain requirements and implementation technologies” (Paasivaara, et al., 2008). According to Shrivastava and Date (2010) it’s not suggested that works are distributed according to the location while assigning a specific responsibility to just one location. He believes that Works should be distributed regardless of location like a normal distribution, which happens in agile methods. Distributing the work according to locations results that teams specialized in particular components and for completing a user story, the works of one or some teams are required, which makes handling user stories much harder (Miller, 2008).
Distributed Scrum of Scrums: Cross-functional teams that are isolated across multiple locations and meet regularly are the characteristic of this distributed method. The dependencies between teams are managed to be as lowest as possible and the link between teams are achieved by Scrum-of-Scrum meeting where Scrum Masters of teams join it (Lee and Yong, 2010; Sutherland, Schoonheim, Kumar, Pandey and Vishal, 2009). It’s appropriate for newcomers to agile development (Sutherland, et al., 2007).

Fully Distributed Teams (Integrated Scrum): Like distributed Scrum of Scrum method, teams are cross-functional but they’re dispersed across sites and members of each team are located in multiple sites (Lee and Yong, 2010; Sutherland, et al. 2009). Daily Scrum meeting could solve the communication and coordination burdens, which is useful to break down cultural barriers. It could achieve a similar performance like a small, collocated team if well implemented (Sutherland, et al., 2007).

3.4.2.1.4 Yahoo! Case

To understand how teams are distributed and to be able to compare it with the situation in Volvo in the empirical study, My Yahoo! Project is explained. It’s an easy to use custom homepage with highly customizable features, developed by distributed agile method. The development teams were in Sunnyvale, USA and Bangalore, India and the international teams, responsible for localizing the product, were working in different countries in Europe, Asia Pacific, and America. In Sunnyvale, they had representatives from different stakeholder groups like product management, marketing, engineering, QA, etc. and each group was empowered to take decisions on behalf of their group. Two special rules have been assigned for this case. One chief Scrum Master, a Scrum Master for facilitating the whole process, is used however project manager often took this role. Also, an international Scrum Master facilitates the process of localization and collaboration between the global product team and international teams while both mentioned teams working with the global product team in Sunnyvale. Finally, local Scrum Masters were assigned for each region. The local product managers and engineers traveled to Sunnyvale for the product kick and worked with the global product team through the international Scrum Master for two weeks (Lee and Yong, 2010).

Figure 4 captured from Lee and Yong (2010) shows how we can divide the Scrum teams into core team, international Scrum teams, regional and local teams:
3.4.2.1.5 Scrum Meetings

Holding various meetings is one of the main features of Scrum which suggests structured way of communication. Having an optimized Scrum meeting is possible by considering some issues about it. First of all, the location of different members is important. Achieving strong central control of teams across geographies is possible by centrally locating Scrum Masters, Product Owners, and Architects. A Product Owner which are located in the main site, could be assigned to each team. A chief Product Owner meets regularly with all Product Owners to assure coordination of requirements (Sutherland, et al., 2007). The same could happen to Scrum Master to have one chief Scrum Master and local Scrum Masters in each location (Lee and Yong, 2010).

Two other problems and issues in distributed Scrum meetings are that more time is usually needed to complete these meetings and less time is available because of time-zone differences. One solution when the overlapped time is less is to divide it into some shorter meetings, which fits well for all sites (Deemer, et al., 2010). Also, for meetings, a time should be chosen which is inconvenient for one side or the other, and rotate it between sites for reducing the stress and increasing the productivity, especially when the time-zone differences are high. It could be changed every one or two weeks. If it’s not possible, the meeting should be organized in each team location, a convenient time for them, but a member is needed to write-down the report and summary of the meeting and sends it to another member of the other team to read it aloud before starting the meeting at the other site (Sutherland, et al. 2007).

**Sprint Planning Meeting:** It’s a meeting organized before each Sprint to plan it based on the Backlog. It’s when the team demonstrates the new functionality to the Product Owner, the customer, or the interested parties (Paasivaara, et al., 2008). This meeting is divided into two sections. The first part of it is for discussing about
requirements or user stories and the team’s feedback regarding user stories and its goals. This part is considered for answering ‘What’ questions but the second part focuses on ‘How’ questions. Here, the development team tries to identify tasks from the previously chosen stories and deduce how much, in hours, it will take them to turn these tasks into potentially shippable product increments. Usually for this step a task board is used which could be a white board (Pham and Pham, 2011). Requirements come from what we know as user stories. Some of them are lengthy and have many details, which bring questions from other locations teams and could be solved by instant messenger or e-mail. Requirements from user stories are broken down into development tasks for developing (Sutherland, et al., 2007).

For a DSD team with high time-zone differences, it’s suggested that Sprint Planning meeting is divided into three phases: distributed meetings, local meeting in onshore and local meeting in offshore locations. First, Product Owner should have meeting with each site to prepare the Sprint Planning. After receiving the explanation of user stories, they meet jointly to create tasks. Distributed meetings should be during synchronous working hours for explaining Backlog items and is followed by consecutive site-specific parts. It’s recommended that after that the onshore team does the meeting and the next morning offshore team that could be opposite based on the locations of the teams. The challenges are that time zone differences make it difficult to arrange longer meeting and cultural and language differences caused silence of some participants. Also, if teams are not using video, it’s difficult to recognize speakers (Paasivaara, et al., 2008). After This meeting, the actual Sprinting work is started along Daily Scrum meetings, or Daily Stand-up, which is discussed in the next section (Pham and Pham, 2011).

**Daily Scrum Meeting:** Process management and synchronizing work between different sites is possible by organizing proper Daily Scrum meeting (Sutherland, et al., 2007). If the whole team participates, pre-work is needed with customers and developers to clarify as many issues as possible. Sutherland et al. (2007) suggested answering three Scrum questions before the meeting:

- “What did you yesterday?
- What do you plan to do for today?
- What impediments are getting in your way?”

Pries-Heje and Pries-Heje (2011) recommend the forth question to ask about ‘innovations’.

Answering these questions before distributed meetings are helpful to minimize the time needed to join teleconference and overcome language barriers. The answers should be sent by E-mail throughout the project teams to have a better and smoothly meeting. Daily Scrum meetings help members to get to know each other and provide a coordination mechanism for members (Sutherland, et al., 2007). It’s also helpful for improving cross-team project management and revealing the possible problems early on (Jansen and Zilmer, 2003; Danait, 2005). It’s usually 15-30 minutes distributed meeting. Mainly, developers, Scrum Master and team’s Product Owner join in one meeting room preferably using teleconferencing and web cameras on the common working hours. A research shows that in the first days, the meeting was just couple of minutes but after some days, teams learn to convey useful information (Paasivaara, et al., 2008). Unlike the traditional method, the team gathers every day to inspect and checking the team’s progress toward the spiritual goal and not tracking project status
(Pham and Pham, 2011). This practice is more useful for smaller teams and fully distributed Scrum practices discussed in section 3.4.2.1.3.

For some teams, there is no Daily meeting that whole team members participate in it but they follow another practice, which is more suitable for distributed Scrum of Scrum teams, discussed in section 3.4.2.1.3. In this structure, local sub teams have a Daily Standup meeting at the beginning of the day. After that, Scrum Masters meet in a Scrum of Scrum meeting occasionally, every day, some days in a week or once in a week depends on the structure of the project (Sutherland, et al., 2007).

Sutherland, et al. (2009) suggests using proxy for teams working with large time differences, where it’s impossible that all members of the both teams join the meeting. The proxy should join standup meetings in onshore and offshore sites. It’s recommended that the proxy is located in the main location, because of available knowledge on that site. For this practice, it’s needed that the proxy changes his/her working time to the evening and the offshore team need to start their work earlier or finish it later. Different people in onsite location could have the role of proxy so interaction with various members is experienced. For these cases, once a week, the whole team joins a meeting to see each other and discusses about related issues (Sutherland, et al. 2009).

**Retrospective Meeting:** After the Sprint Review (will discuss in next section) or prior to the next Sprint, the team goes through a Sprint Retrospective to discuss about things worked and didn’t work during the current Sprint. The goal is to have a more effective collaboration for the next Sprint (Pham and Pham, 2011). At this meeting, the processes, practices and the challenges are discussed and if needed, the way of working is changed before the next Sprint (Paasivaara, et al., 2008). Pham and Pham (2011) suggests having three hours meeting for a monthly Sprint that is adjustable for other time-boxed Sprints. Like Sprint Planning, Retrospective meeting is beneficial for improving communication, coordination, and team cohesion in a distributed project (Holmstrom, Fitzgerald, Agerfalk and Conchuir, 2006). Three questions are suggested for this meeting while teams, Product Owner and Scrum Master participate in it:

“What have been good during the Sprint?
What has not been that good?
What kind of improvements could we do?” (Paasivaara, et al., 2008)

The challenges of this meeting for distributed teams are more since more issues are required to be solved. As discussed before, shorter Sprints give more opportunity for inspect and adapt cycles, which is useful for understanding the problems sooner. In addition, having Product Owner in this meeting is important since many dysfunctions cause problems for collaboration between Product Owner and the team (Deemer, et al., 2010; Pries-Heje and Pries-Heje, 2011).

**Sprint Review:** Before ending any Sprint, team members with Product Owner gather to have a Sprint Review organized by the Scrum Master. Pham and Pham (2011) recommends four hours Sprint Review meeting for a four-week Sprint or two hours for a two-week Sprint. One objective of these meeting is discussing about what is done and what was not done, which is useful for both team and Product Owner. Another part of this meeting is for demonstrating the progress and what was done to
the Product Owner by the team. Product Owner is also required to give feedback about the product and inform team about any update and change related to the product or market direction (Pham and Pham, 2011).

**Sprint Demos:** It’s arranged as a distributed meeting and team members, Scrum Master, and Product Owner participate in it. The agenda for demonstration is prepared in this meeting. Benefits of it include increasing transparency, preventing the problems by providing a monitoring opportunity, and ensuring that requirements are clear especially for the offshore team (Paasivaara, et al., 2008).

**Release Planning:** In Scrum, Release Planning with Product Owner and the team is suggested however it will be difficult because it needs prior information about the product before the Planning meeting (Paasivaara, et al., 2008).

### 3.4.2.2 Time-zone Differences Challenges and Practices

The problem of time-zone differences is more problematic than the geographical differences since significant time-zone difference cause serious communication blackouts. This problem becomes important when you need someone in other locations but that member isn’t available. The time-zone differences between three to four hours are acceptable since teams could have meeting and pairing working on the overlapped time (Miller, 2008).

If the time-zone difference is large for an offshore team, a representative is needed to be active in the meeting discussion and pass the results to the onshore team. The representative should have worked with the onshore team and has a good rapport with them (Shrivastava and Date, 2010). He should also have a core role in the offshore team, participates in team’s discussions and meetings and be available to do his job. The representative conveys the work and required information, which has captured from the work of the offshore team. Following this practice has some problems since the practice is like delegation and teams don’t have the sense that they work as one team. Also, the problems of using proxies for transmitting information, discussed in Product Owner section of section 3.4.2.1.2, occur for it. On the other hand, it’s much harder to ramp up a new team since they are completely disconnected from the onshore team (Miller, 2008).

There are some practices for team problems with less time-zone difference cases. The synchronization of working hours is a widespread practice in these distributed structures. Constant communication is possible only through the maximization of overlapping work hours. For example, early morning shifts for the onshore site and late evening shifts for the offshore site (or opposite, depends on where teams located) let the teams to ‘share the pain’ of synchronizing the work schedule (Ramesh et al., 2006; Paasivaara, et al., 2008). Phalnikar, et al. (2009) believes that testing and deployment teams can work separately so when it’s needed to change the working hours for increasing the overlapped time, they can run on a separate schedule. Because it’s recommended that the whole team participate in the Planning or Retrospective meetings and because of the time-zone differences between locations, it’s necessary to find “a sufficient block of time which fits all parties” (Layman, Williams, Damian and Bures, 2006). On the other hand, the non-verbal formal
communication of DSD is useful for overcoming the problem of non-overlapping working hours (Shrivastava and Date, 2010).

A function-focused team is another practice to solve this issue but as discussed before in section 3.4.2.1.3, it’s similar to isolated team structure and will cause mentioned problems. Also, while in agile, teams focus and work together to solve a problem and do a job; in this practice, they push the problems back and forth between themselves, which decreases the functionality of the team. Distributing teams adds the problem of a functional team (Miller, 2008). One practice is increasing the balance of teams and having a mirror of roles by hiring people with the same responsibility as needed in the offshore or onsite site. This role could be an Architect or a Development Manager. This practice is opposite with what we have in function-based teams and is useful to decrease the dependency between teams and avoids teams’ deadlock because of a needed specialty or functionality. These members with same roles could work together to achieve better performance and collaboration (Paasivaara, et al., 2008).

### 3.4.2.3 Communication Challenges and Practices

Collocation reduces project risks and collocated teams were also 20 percent more successful that geographically distributed teams, according to a survey of agile projects (Miller, 2008). The main reason that teams prefer to be collocated is to communicate. Software development is possible by communication between people who are involved; Customers discuss about what they need and gives feedback, members discuss with each other to build the product and with customers to receive feedback and satisfy them more. Communication facilitates collaboration and knowledge sharing between teams. More poor communication means wasting more time, which has negative impact on value, quantity and accuracy of the end result. Also, The best method for communication is face-to-face when facial expression, body language, intonation, and words are conveyed (Phalnikar, et al. 2009; Deemer, et al., 2010).

There are some main differences between agile and DSD regarding communication. Distributed development causes some problems for teams’ communication. The main problems consist of lack of close physical proximity, unsatisfactory team cohesion, insufficient shared context and knowledge, and unavailability of team members. Other challenges are difficulty to initiate communication, decreasing the frequency of it, misunderstanding and misconception, and also increasing the communication cost, time, money and staff. The impact of reduced communication on coaching is another challenge (Ramesh, et al., 2006; Paasivaara, et al., 2008; Shrivastava and Date, 2010; Miller, 2008). On the other hand, agile emphasizes on casual conversation between team members and ignores detailed specifications, which is used more for waterfall approach. Agile also asserts that intensive person-to-person communication between team members and also with the customer is necessary to solve problems early on while distribution causes team’s dysfunction by inhibiting communication. In addition, Agile is different with DSD that relies on formal mechanism of communication and explicit documentation and the success of agile relies on intensive face-to-face communication, which is also a part of agile principles, and if it’s not possible to have it, it must be replaced with other approaches like weakened practices for communication (Ambler, 2008; Miller, 2008). As a result, finding a balance
between them for communication in a distributed agile project is required (Miller, 2008; Ramesh et al., 2006).

To have an effective agile distributed team, communication barriers must be overcome both by commitment from team members and also by using tools to support communication (Shrivastava and Date, 2010; Ramesh et al., 2006). Using several different kinds of communication media and multiple communication methods that can be applied in parallel is helpful for improving communication. Individual and conference telephone, teleconference, videoconferencing, e-mail, instant messaging, Wiki, application for sharing software and desktop sharing are some common approaches to substitute face-to-face communication (Paasivaara et al., 2008). Conference calls and online communication is a major way of interacting between the global product teams and each local team. Instant messenger is used for synchronous communication and e-mail for asynchronous communication (Shrivastava and Date, 2010; Miller, 2008).

Using E-mail has some advantages and disadvantages. While for using it, there is no need that both parties be available simultaneously and the discussion is recorded for later accesses, much more time is consumed and more effort is needed. For example, ten mails back-and-forth are sent instead of five minutes talking. Also, misunderstanding might happen because of missing mood, tone and intent of the writer. A live communication tool using audio and video is an effective substiation for e-mail as a primary way of communication (Deemer et al., 2010).

Audio-only meetings have also some problems. While the facial expression and body language are missed, the source of the voice is sometimes hard to be identified. Also, the natural flow of meeting is sometimes missed by unintentional interruptions. In contrast, sometimes participants don’t talk because they’re afraid to interrupt. On the other hand, if people have different and unfamiliar accents, it’s much harder to understand them in a situation that you couldn’t see their face. Finally, people are more likely to be ‘multitask’ like surfing the Internet, doing their daily jobs, etc. and pay partial attention to it while using audio-only meeting. Using video, gives a sense that people you are talking with are not just a voice, instead a real and live human being (Deemer et al., 2010).

In general, increasing the communication bandwidth is important to build a successful distributed team. Minimizing the time for setting up a meeting across locations is possible by having a conference phone and projector, which is easily accessible preferably in the team room. In addition, web conferencing tools with projector for sharing application or using video conferencing speed up and facilitate the communication. They could also be used passively to see the room and find out who is available. Deemer et al. (2010) suggests that necessary equipment are wide-angle (which makes it possible to see more people), high-resolution webcam and high quality conference phone connected via a landline. Finally, However sometimes it’s enough to use Skype but landline gives more confidence and quality.

Tools for facilitating the development process are also required. Issue tracker tools are helpful for both maintaining documentation and proper transparency and also tracking project status. It’s used to report issues and assign priorities to the tasks. In these tools, Product Owner, project manager, or Scrum Master updates separate Backlogs
for each team. These updates are accessible by all members but each team member, based on his/her responsibility has different level of access, like reading, updating, picking items and monitoring its progress (Shrivastava and Date, 2010; Paasivaara, et al., 2008; Sutherland, et al., 2008). Wiki is another tool useful for documentation and sharing information globally. It’s used by members of teams to update what they did daily and what impediments they experienced (Sutherland, et al, 2009). Using Wiki and message board instead of e-mail are useful for saving time (Phalnikar, et al. 2009). Smart board, which is a computerized whiteboard, is recommended and however they are regarded as costly equipments, it offers a better experience for discussing the architecture and having Scrum meetings. A shared mailing list, single continuous build system, and a single code repository, suggested by Sutherland, et al. (2008), are some other solutions and practices to overcome the distance problem and facilitate development and communication.

3.4.2.4 Building One Cohesive and Virtual Team

A problem in distributed teams is that teams often act independently and the collaboration and communication between teams doesn’t work properly even by using proper communication tools. Also, cultural differences exist which might cause problem for team coherency and collaboration. As a result, building one cohesive team, which team members have the feeling that there is one team in general and all of them are working for one goal, becomes necessary for teams to work smoothly (Sutherland, et al., 2008; Deemer, et al., 2010). Building team particularly trust between members is possible by focusing on team morale and personal relationship between teams. Without it, misunderstanding and miscommunication occur on the mistakes caused by members of the members. It’s usually the responsibility of Scrum Master and project manager to build this kind of environment (Sutherland, et al., 2008; Lee and Yong, 2010).

In a collocated agile practice, it’s suggested that members have informal communication to increase the coherency of the team but in a distributed setting, teams couldn’t meet frequently so different methods are required for team building in a distributed agile structure. These methods consist from frequent visits using by distributed partners, visits by senior managers to establish agreement in the beginning of the project, building cohesive team culture by utilizing members who worked before with other teams till frequent and various ways of online communication. I discuss more about these practices in detail in the next paragraphs (Lee and Yong, 2010; Ramesh, et al., 2006).

Before starting the project and for the first iteration, it’s suggested that teams have face-to-face meeting, which is useful for building trusts and creating rapport. Social gathering before the project is a practice for team members to discuss about issues not related to the work and to know each other. This gathering is also used for discussing about future actions and having a team exercise and it could be arranged several times during the project. It’s useful because the whole team participates in a meeting when the critical decisions are made so the members will be more committed to the decisions (Miller, 2008; Paasivaara, et al., 2008). Having the whole team in one room for the first iterations is also helpful for creating personal bonds. Also, by arranging it, both clients and developers can get acclimatized with the processes, tools and the
application and they can work together to finalize requirements and develop a high level architecture. In addition, business cycle and needs can be regarded in the first iteration where all members exist. Some project managers consider these practices costly but they give the opportunity to team members to get to know each other, getting a better picture of the project, increasing trust and speed of developing the project so it should be considered as a proper investment (Ramesh, et al., 2006; Paasivaara, et al., 2008; Miller, 2008). Paasivaara, et al. (2008) categorized the meetings, which occur early in the project to build a relationship as ‘seeding visits’, and believed that they should be intense during the early development cycles.

Not limited to the first iterations, offsite team members need to have face-to-face communication with onsite experts discussing about difficulties and asking questions. As a result, frequent visits are necessary to build and maintain relationships between distributed team members. It’s suggested that both offshore and onshore members visits. Customers and product managers should visit the development team and developers or developer representatives should travel to the customer site. Also, It’s suggested that members collocate in one location in the critical phases of the project like before a release or before the last release and finalizing the product. One benefit of bringing the team together for the last couple of iterations is to make releasing the final version of software smoother (Paasivaara, et al., 2008; Shrivastava and Date, 2010). Paasivaara, et al. (2008) categorized these Shorter visits which are useful to maintain the collaboration relationships as ‘maintaining visits’.

Understanding the way things work in other sites and how that members communicate and deal with different issues are important for successful collaboration. There are some ways to handle it. The ambassador practice is a practice used for cultural, development and business-oriented exchanges. The ambassadors travel to the other sites and take part in Daily and Retrospective meeting of the other sites and by receiving some knowledge; more effective future collaboration is possible. Rotating team members in a way that at least one member be away from his/her main location is another useful practice for distant teams for maintaining trust, getting along better and developing peer-to-peer relationship. As another option, rotating gurus, who are responsible for initial training and monitoring the other sites, is a useful mainly for larger budget projects (Paasivaara, et al., 2008; Nisar and Hameed, 2004; Hogan, 2006). Finally, sending experts to work with offshore team is recommended. For a distributed practice between Norway and Malaysia, explained by Paasivaara, et al. (2008), two experts from the onsite location, one for running the office and another one expert in the customer and business side of the project moved to offsite location.

One problem when multiple teams are working together for one project is that some teams might form cliques by location. If there is one onshore team and the team has more access and communication with the Product Owner, they will have information advantage over the offshore teams, which will cause a wedge between the two groups of developers (Deemer, et al., 2010). The possible solution for it is to make a balance between sites by sending business side people and experts to other location as discussed above.

The problems of cross-cultural communication sometimes happen because of differences in work style of the primary and outsourced sites (Sutherland, et al., 2007). Sometimes team members especially in offshore sites are afraid of telling the
problems to the Product Owner, Scrum Masters or other members in a Retrospective meeting, etc. and they don’t ask questions because they are afraid that it might be conceived as being incompetence. Cultural differences in reporting impediments especially for Asians is another issue as they don’t tend to talk about impediments so it’s necessary to encourage them to be more direct. In Xebia ProRail PUB case study for Dutch railway, explained by Sutherland, et al (2008), cultural differences between Dutch and Indian members of the Scrum distributed team is explored. While Dutch members of the team were more direct, Indians were indirect, careful in their expression. Also, the degree of hierarchy is higher in India rather than Netherland. These differences also exist between other cultures and the experience of Scrum Master helps to solve it. He must act in a way that he is looking for both good and bad news and no one will be punished for honesty. By encouraging everyone to talk, it’s possible to get an overview of the project situation and makes it easier to monitor the offshore situation. Problems will also be recognized more quickly because it’s more difficult to hide them for a longer time. Other benefits are increasing transparency to the other site and enhancing communication across sites (Paasivaara, et al., 2008). These problems required to be regarded before it becomes an obstacle for team progress.

Finally, there are some technical practices that encourage member to collaborate more, which is useful to build a coherent team. First, continuous integration as a technical approach is useful to reduce or solve configuration management issues. The solution is to integrate the codes and build the entire system whenever the changes were made. It eliminates the huge integration problems, which might occur at the end of projects by considering them early on (Phalnikar, et al. 2009). It’s also useful for members to collaborate more, and for them to assure that no one else is interfering with his/her piece of code. Pair programming that members from different sites working on the same piece of code has social benefits that members are collaborating while coherency of team is increases (Sutherland, et al., 2007).

3.5 Theoretical Research Results

1. What issues need to be evaluated before starting to use distributed agile software development method?

Agile is a trendy way of developing software, which has started to be used in many projects. The main features of agile are iterative development and involvement of different stakeholders like customers and users in different phases of project. These features are important to response to the increasing rate of change in software development and business environments. Agile also promotes intensive communication between members preferably face-to-face to have a better collaboration between members (see chapter 3.4.1.1.2). Saving time and cost, more productivity with fewer people and easier measurement of the project progress are considered as other benefits of agile (Pham, and Pham, 2011; Shrivastava and Date, 2010). On the other side, evaluations on customers, team members and cost are necessary before starting to use agile. If the customer has some business goals but not didn’t provided detailed requirements and wants to participate in the different phases of project like gathering requirements; agile is suitable for it to response to the uncertainty of the customer (Sutherland, et al., 2007; Tudor and Walter, 2006; Khan,
In addition, agile is a cost effective approach as it focuses on producing what customer needs from the beginning of the project but other costs like cost of user involvement and cost of training should be considered for projects. Finally, the success of agile method depends on the high-skilled developers and members (see chapter 3.4.1.1.3).

Distributed software development has also its benefits and challenges. The main factors that encouraging companies to use DSD are accessing global markets, using global talent, reducing costs and increasing productivity (see chapter 3.4.1.2.3). Before starting a distributed project, the level of distribution should be selected based on available resources and goals of the project. There are two different ways: partial offshoring that business team are near the customers and working closely with them and complete offshoring that the business team and development team are both offshore and work remotely with customers. Each of them has some benefits and problems (see chapter 3.4.1.2.2). On the other hand, some social, cultural, management and technical problems occurs in DSD. The problems include cultural issues, inadequate communication and problems in management and transferring knowledge. There are also some challenges for strategic issues on managing the work and assigning decision responsibilities and managing the work across location. Finally, there are some issues with integration of the work of different teams and also for documentation (Shrivastava and Date, 2010; Sutherland, et al., 2007; Lee and young, 2010; Phalnikar, et al., 2009) (see chapter 3.4.1.2.4).

Agile could bring some benefits to a DSD project. First, because of incremental and iterative development, effective suggestions and solutions from both customer and team members are considered and the possible faults are ignored early. Also, the visibility of the project is increased. Communication is another factor and because of required continuous integration for the project and because agile emphasize on intensive structure way of communication, the level of it increases for a DSD project (Shrivastava and Date, 2010; Ramesh, et al., 2006). Finally, involving customers brings the benefits that requirements are better understood and the transparency of the project increases for all partners including customers (see chapter 3.4.1.2.5). More details about challenges and efficient practices of combining agile and DSD are regarded in the next question.

2. What are the main challenges while using distributed agile method and which practices are needed for coping with these challenges and problems?

The main distributed agile challenges are categorized into four parts: How to have a better distributed Scrum practice, How to overcome time-zone difference problems, how to solve communication problem and finally, how to build one cohesive and virtual board.

The first challenge discusses about how it’s possible to have an optimized distributed Scrum practice. Scrum is a trendy agile method, which emphasize on working simultaneously on requirement, analysis, design, and testing for each iteration and delivering the software at the end of it. Iterations in Scrum are limited into Sprints. Longer Sprints give opportunity to produce more functionality and decreasing the frequency of Scrum meetings but in longer sprints, it’s possible to produce more
wrong functionality (see chapter 3.4.2.1.2). Teams in Scrum are working in development teams and business team. In development team, the role of Scrum Master is important since he is responsible to guide the team to use agile way of working. He should also work closely with developers to connect them for the problems and organize needed Scrum meetings. Product Owner also has important roles in team as he is responsible for business values and represents the customer. Product Owner needs to have an extensive collaboration with both customer and development teams. Some teams are using proxy roles especially for Product Owner in distributed location (Miller, 2008; Scharff, 2011; Pham and Pham, 2011). There are some problems with using it since the proxy doesn’t have equal knowledge as the main role and misunderstanding might happen while conveying the information. Also, using proxy slows down the speed of the project (see chapter 3.4.2.1.2).

Sutherland, et al. (2007) categorized three different Scrum team structures that choosing the appropriate one based on the need of the project is important. First is isolated Scrum structure that teams are not cross-functional and are isolated across locations. Some problems happen with team communication and relationship in this structure while teams become specialized in particular components. It’s not adjusted well with novel projects having uncertain requirements. Second structure is distributed Scrum of Scrum teams, which are cross-functional teams isolated across location but members meet regularly. It’s appropriate for projects that require low dependencies between teams. Finally, fully distributed teams that teams are cross-functional and dispersed across location. This practice is useful to solve the communication and coordination burdens (see chapter 3.4.2.1.3).

Scrum meetings are important parts of Scrum that suggest structured way of communication. Distributed meetings should be in a time, which is convenient for two parties. If it’s not possible because of time-zone differences, choosing a convenient time for one team and rotating it for the next weeks for other teams are required. The location of specific members is also important which depends on the level of distribution. Also, using video is suggested for the distributed meetings (Sutherland, et al., 2007; Lee and Yong, 2010; Deemer, et al., 2010). There are some meetings suggested for each Sprint. **Sprint Planning** happens before each Sprint for discussing about Backlog and it’s recommended that all members participate in it. If it’s not possible for team working in large time-zone differences because of small overlapped time, teams could have distributed meetings followed by site-specific meetings. Another important meeting is **Daily Scrum meetings**. Some teams prefer to have distributed meetings between all members of development teams. Because the number of members is usually too many that all members be able to discuss about their issues in 15-30 minutes, Pre-work on answering three Scrum questions and sending them by mail are required to efficiently use distributed meetings. For bigger teams and for teams with lower level of dependency, the practice is that each site has Daily meeting in its location and them Scrum Master of teams have Scrum of Scrum meetings, daily or weekly. Another practice for teams working with large time difference is to use a proxy who joins both meetings. The proxy should change his working time to be able to join both meetings. **Retrospective meeting** happens at the end of each Sprint to discuss about what worked and what didn’t work during the current Sprint to have a better practice for the next Sprint. It’s recommended that Product Owner joins this meeting. **Sprint Review** meeting is also for the end each Sprint, discussing what was done and what was not done and demonstrating the
progress while **Sprint Demo** meeting is for preparing the agenda for demonstration (see chapter 3.4.2.1.5).

Time-zone differences cause another challenges for teams. This issue is more related to unavailability of members for the needed dependency and causes problems for communication and collaboration. The synchronization of working hours to increase the overlapped time is a useful practice to mitigate this problem. Also, as discussed in Scrum meeting section, arranging a sufficient block of time for important meetings that works well for all parties is necessary. For teams with large time-zone differences, the representative or proxy practice is used that the representative joins offshore meetings and pass the results of meetings to the onshore team. This practice is similar to delegation and causes problems of using proxies. A function-based team is a practice for decreasing the dependency between teams but its similar to isolated team and will cause the mentioned problem. Also, it provides a competitive environment that teams push the problems back and forth between themselves (Miller, 2008). Finally, increasing the balance between sites and having a mirror of roles in offshore locations is useful for dependent teams (see chapter 3.4.2.2).

The third challenge is communication and it occurs because teams aren’t collocated and they aren’t able to collaborate and share the knowledge in a proper way. Finding a balance between agile, which emphasizes on casual and intensive person-to-person communication, and DSD, which more focuses on informal communication, because of lack of physical proximity, is necessary to have a better practice (Miller, 2008; Ramesh, et al., 2006). Using different kinds of communication media and methods and in general increasing the communication bandwidth are recommended to overcome DSD communication problems. Instant messenger is used for synchronous non-verbal communication and E-mail for asynchronous communication. E-mail has advantages that for communication, there is no need that both parties be available and a record of discussion is available for later access but more time is consumed and misunderstanding might happen because of missing the mood. Using video is recommended for verbal communication since in audio-only meetings, expression and body language is missed, interruptions occur during the discussion and members are more likely to be multitask (Deemer, et al., 2010). Videoconference facilities in rooms to be used both passively and for meetings are useful. Finally, proper development tracking tools both virtual and physical board in addition to using wiki for sharing information globally are the needed tools for facilitating development procedures (see chapter 3.4.2.3).

The final part of challenges discusses about the ways to build one cohesive and virtual team. The problems are related to cultural differences and tendency of distributed teams to work independently and the solution is to build team morale and personal relationship. Different practices are required to reach this goal. It recommend to start team building practices before starting the project and for first iterations by gathering the whole team in one place for discussing about issues not related to work, making critical decisions for the project and getting familiar with processes and tools. After that, frequent visits are needed to maintain the relationship. Members from both onshore and offshore locations should travel especially in critical phases of the project. Ambassador practices, rotating members between teams, sending experts to work with offshore teams are other practices to build the team and make the balance between teams (see chapter 3.4.2.4).
4 Empirical Study

The empirical study covers the data I received from my interviews with seven different members in Volvo who are active in distributed agile projects.

4.1 Purpose

The main purpose of this section is to verify the findings of theoretical study. By using interviews, I aim to receive an evaluation about problems and challenges of using distributed agile method in Volvo IT and find out about the strengths and weaknesses of practices used to cover the challenges.

4.2 Interviews

I had seven interviews with people who are currently working as part of Volvo IT distributed agile projects. My first interview was in Volvo IT office in Gothenburg and it was more about the Volvo IT distributed agile project guideline and the current situation of the projects. I used online communication methods like Skype and live meeting for my other interviews. The duration of each interview was around 60-90 minutes. Before each interview, I evaluated the role of interviewee and based on his/her responsibility, I prepared some questions and sent it by E-mail. By using this approach, since I covered the project goals in interview questions, the interviewee had an evaluation about my project and was prepared for the interview. I prepared two versions for each interview. Another version was a detailed version for myself to be ready for deeper discussions.

Before starting each interview, I gave the interviewee an evaluation about the project and my background. After that, I covered my questions by considering the point that I followed semi-structured interview to be able to capture more detailed data from interviews.

Interview Questions:
Below, you can find an example of main questions that I was sending to interviewees:

First of all, I would like to know about your education and work experiences background?

What’s the specification of your current project? What’s your role and who is your customer?

What’s the structure of the team? How different members like architects, business analyst, project managers, instructors, developers, testers, etc. are distributed in different locations?

How the teams are built? What are your criteria for choosing members?
What are the main reasons and forces for using distributed agile?

Suppose Volvo is decided not to continue agile distributed projects, what do you show as a proof that it’s a wrong or a good decision? Could agile be used for all kinds of projects or based on project specifications, agile is chosen?

How the problem of time-zone differences is mitigated?
Have you ever experienced a project that teams are working with high time-zone differences? What practices did you follow?

Have you ever had a situation that you need to discuss with a distributed member about an issue but he/she isn’t available? What was the problem and how did you face with this problem? Describe an example?

What practices are followed to have the feeling of being in just one team for a distributed agile project?

Which practices are experienced to get to know the member of other teams (more personally) and their cultures? Are they enough?

How is the frequency of travelling for different members of teams?

What’s the scope of communication with teams in other locations?

Which meetings are experienced for each Sprint?

Are tools satisfying you? What are the problems with tools that you’re using for communication and sharing the work progress?

How frequently you use video for the distributed meetings? What are the reasons Volvo isn’t encouraging members to use video?

What’s the main work style difference between your location and other locations? Please explain specifically for each location that you have experienced?

4.2.1 The First Interview

The interviewee is working in Volvo IT in Gothenburg, Sweden and has been with agile methodologies for seven years. In the projects, he was mainly acting as a Scrum Master. He has participated in two major agile adaptions in Volvo; one was Scrum and part of XP programming for six years and the other one is mapping Scrum more on XP. Also, he is analyzing the opportunities for bringing more Kanban into projects.

In this interview, the interviewee explained the current situation of Volvo IT distributed agile projects; the challenges and opportunities. He also explained parts of guideline that he prepared with help of Naresh Jain, one of the leaders of distributed
agile development. He claimed that he used this guideline for the first distributed agile project between Skovde, Sweden and Wroclaw, Poland. That project was almost full implantation of needed Scrum practices like team building activities, co-locating of all members for the first weeks, having seeding visits, etc., reach the purpose of building interpersonal relationship and common way of working. They did the same approach for PBS project. For beginning of that project, all of team members, testers and developers (10-12 people), from Poland every three week came to Sweden however he believes that it’s highly unusual and not like a normal project.

The guideline that he discussed about is divided into two sections: Patterns and Anti-patterns. Patterns discuss useful principles, values and practices that are recommended for distributed agile projects. It emphasizes on different issues like empowering team and its members to take decision about the tasks that they are directly related, trusting members and also considering no difference between members in different locations. He believes that agile projects should have small, cross-functional teams. Also, the guideline explains the importance of traveling for all members. It consists of discussion about seeding visits and team building practices at the beginning of the project and ‘cross pollination’ that all members from different locations should travel in different phases of project to transfer the knowledge and culture.

Regarding needed patterns of communication, he discussed about providing an environment that all members are comfortable to communicate. He believes that it’s the role of Scrum Master to make sure that communication and collaboration works well between members and to encourage them to communicate more. The tools that are used for communication in Volvo are Live Meeting, chat, and videoconferencing. The issue tracking tools are physical board and also virtual boards like TFS for .Net Development and Jira for Java development. He believes that “using a tool like TFS is not that much necessary, the only thing we need is a board with a little bit imagination. It could happen by updating the board, take a snapshot and sending it by mail, etc.”

The anti-pattern part of the guideline suggests not to use proxies for communication and recommends to “talk to the person you need to talk to immediately”. Proxies lead to low bandwidth communication, depending on few individuals and misunderstanding of the issues and problems. The anti-pattern also discusses that there are some wrong expectations about agile that it’s possible to save a lot on cost by using it. Another wrong expectation is that since you’re using agile, it’s possible to change everything but the interviewee believes that it’s more challenging in agile to change. Finally, the guideline suggests avoiding giving huge requirements to the offshore team while expecting to have a delivery after some months as by using this approach, bad results are received.

4.2.2 The Second Interview

The interviewee, an employee in Volvo IT in India, finished his bi-mechanical engineering in 1987 and then shifted to IT. His agile experiences were started in 2007 where they had a project in another company, which needed to be delivered in a very quick time to the US market. He joined Volvo IT in August 2010 and handles teams
in India and manages projects. In Volvo IT, he was teaching Scrum to new comers in two days courses. He was also coaching multiple project teams, which were in need of better return of investment like teams that need to become more dynamic following Scrum methods and also teams that are facing challenges using Scrum methodologies. Last year (2012), he got trained Kanban and now he is training both Kanban and Scrum.

He claims that the main force for Volvo to have distributed projects is “that Volvo is a global company and we need global people across the world and in different and multi sites” so the need for Volvo to be in different countries is mainly for local support and local language and local culture. The cost is another factor for distributing to some countries. Considering different roles in projects, he claims that in each team, there are roles as agile coach and Scrum Master but the need to have one distinctive Scrum Master who talks with all of them is seen.

He believes that when teams are starting a project, they don’t know each other well and they don’t know how to work more collaborative. The practice used to solve it at the start of the project was travelling members from India and Sweden to the other site for some weeks. Number of members who are travelling and the duration of it depend on the budget of a project and the level of necessity that members be in other locations. This practice repeats after some months and another person moves to other locations. He believes that this cycling travel is helpful for members not to get separated.

Considering collaboration between India and other countries, he believes that in general they don’t have too much problem working with Sweden since the overlapped time is good. Also, when it comes to culture, He claims that they are very much comfortable working with the Swedish culture and also believes that Swedes “appreciate the nature of Indian culture”. He continues that to know more about the cultural differences, “when the project is starting, we have some kind of overall understanding of other cultures and the experience of project manager, leaders, and coaches during the project” helps them to understand it better.

4.2.3 The Third Interview

The interviewee joined Volvo IT in India in 2010, November and is currently working as Agile Coach/Consultant within Volvo IT and helping projects/programs in their agile transformations. His skills include project and program management for agile and Lean processes and methods like Scrum and Kanban.

In the Interview, First of all, he described the practices for teams to get to know each other. The first two weeks is important for teams to get adjusted to agile way of working and to know each other. He claims that “we try to get the people to come here, especially the business people to come here and spend some time with the team, maybe a couple of weeks”. For traveling, they consider the “strength”; if most of the team members are offshore, they don’t travel but some people from onshore location travel during major releases.
He claims that traveling depends on some factors like project budget. Also, if a project is critical and members need to work for a long time, they focus on it. He mentioned that “if the project is small and objective is to get the job quickly, then the focus of project manager will be on doing job quickly and the dynamic of the team won’t be as much as concern for the project manager unless it becomes a problem but for longer projects, it’s required to work on team’s dynamic as well because since it’s a longer project, you have to be more concerned about team’s being stable rather than looking out for new people” so " it depends on project duration and team’s focus”. If travelling isn’t possible because of project budget, they focus on videoconferencing for team building.

He discusses that it’s good to have videoconference since you have at least some sort of visual impact. In Volvo, they use large screen LCD or LED TV with projector. Videoconferencing is used for meetings but not frequently, usually once in a week. Also, it’s not allowed to use video for tools like Live Meeting, because of bandwidth limitation. The way videoconferencing is used is also important. It could be used for both teams’ meetings and also for teams to be more connected using online social gathering.

In the interview, he discussed about tools, which are used to overcome cultural problems. One tool is Globe Smart, which has some particular matrix, categorized for certain points of each culture like open communication, assertiveness, etc., and it gives a general view about a culture. Another way is using a team called ‘culture team’ in Volvo. “They work with the team culture, in general not a specific country culture but it’s more for the culture of values, respecting culture of other countries” cultures and respecting individuals so it’s more for generic culture building in the team”. Not too many teams use this facility. Usually, big projects or projects with problems in hand use this practice.

Regarding development tracking tools, he discussed that they use whiteboard and also tools like Jira and TFS but the need for a virtual board, like a large screen TV, in each room to show the updates, new and urgent issues, etc. is seen. He believes that it much better than virtual tools since you need to go and open the system for the latter.

Finally, he discussed that in Volvo projects, ideas and comments of members and developers are heard. He continued, “If developers have some ideas, by not listening to it, the Scrum team will work but it will not be a good functional team. Hearing ideas is useful and if the idea can’t be implemented but at least the people is getting feedback that why it can’t be implemented and it’s more than enough in that case”. Retrospective meeting is a good way to give ideas because some of developers aren’t comfortable sharing ideas and open for them. He explained that in Retrospective meeting, they pass papers to participants to write their ideas and then they collect all of the papers and read all ideas. As the final point, he believes that time-zone differences is not a regular problem since it’s just three and a half hours differences.

4.2.4 The Fourth Interview

The interviewee is working in Volvo IT in India acting as a senior developer in ALPHA project. He has a post graduate degree in Computer Applications.
The interviewee believes that using distributed agile is cost-effective and India is a location to reduce the cost as well as receiving quality regarding skilled developers. According to him, distributing projects are cost effective because with technology growing on, the cost of travel reduces automatically. The travel budget is restricted for Scrum Masters so developers locating in India don’t travel. As a result, many of interactions happen through Live Meeting, which is just audio and not Video. In some periods, developers and other members from Sweden come to India.

The interviewee mentioned that for each project, the team is not constant and it keeps changing based on the plan for the next Sprint and needed knowledge and skills. Also, when it’s a technical activity, the teams are divided based on different technologies that they are involved like UI. He also explained the way Scrum meetings are handled between sites. The procedure is that development manager, setting in India, has meeting with Scrum Masters and also with testing team and then there is a Scrum meeting between him, Volvo project manager in Sweden and also project manager, business analyst and test manager in India. Also, there is another meeting between project manager, business analyst and the customer. Both of these meetings are daily.

The interviewee believes that there isn’t too much problem for time-zone differences however there are times that they need something and it’s too early in Sweden. He continues that there aren’t also too much cultural problems while working with Sweden since they try to understand each other.

The requirement comes via document and Scrum Master analyzes and picks items for the next Sprint. Based on that, there are some discussions with analyze team and customers, and then the tasks are assigned to the team. The development team have enough power to change the requirement and as the interviewee mentioned “there are many times that in our analysis, we see that for requirements it’s good but technically, it might not be a good solution, then we get back to the requirement team and then we look for better solutions”

Finally, the interviewee explained that there is a need to check agile suitability before using it. The important factors for checking the suitability are divided into three factors. First, checking what really customer wants and whether he is willing to spend some time with the team. Second is evaluating the team and whether the team composition made up people who are not comfortable working with team and if they have time to spend with the team. The third factor, according to him, is checking “the structure of the team and what is the kind of work”, if the requirement is always changing or if it’s straightforward and you need to do a simple task which is already documented.

4.2.5 The Fifth Interview

The interviewee studied Computer Application in her post graduation studies and has six years experiences working with .Net development. She has one and a half year experiences working in Volvo IT, which was mainly for ALPHA project.
As she mentioned, for team building practices in the first weeks, their team had some trainings using PowerPoint presentations, discussing about how to work in agile, how to do planning and how to finish the work. The Project manager, setting in India, was responsible for this training while architect was also coaching about the related issues. The meeting with teams in other locations was by Live Meeting. After five months, some onshore members came to India to give some training and a proper picture of what they developed. They also came to India two more times, each time for two weeks. The project manager was also helping them for releasing, which is about merging codes when the release is going to be promoted to the customer.

She believes that benefits of agile for developers are planning and lack of lagging. He Developers are engaging in planning and there is a board to select and take up things from it. Also, if a developer can’t do a task, she has enough power to return it to the queue of works and the task will go to another person. There is flexibility for the capacity of the amount of the work assigning to the team. Regarding releases, the team has a release usually after 4-5 Sprints because the functionality that they need to give to customer will be ready after that period. The third interviewee, participating in this meeting, believes that it’s “more what the customers wants and he wants a package of functionality to be checked and tested”. If the customer has an urgent issue, they will consider it for the next release, which usually takes around six months. After each Sprint, the team checks the product internally. Finally, developers don’t check the upcoming Product Backlog items and other use cases for the next Sprint but the team leadership level does it.

According to this interviewee, they used videoconferencing for their meetings only once and she believes it’s not required because they’re focused on activity and finishing it. She continues, “It’s not like negotiation and discussion that if you see people, you will receive a better result. It’s more about doing the work and the details of it.”

She mentioned that she doesn’t feel any differences between developers in Sweden and architects in Poland regarding work style. She also believes that however the time-zone differences is four and a half hours, she doesn’t have any problem with it; if they have any issue, they send an e-mail to set up a meeting, after that they work on things that isn’t related for the meeting till they will be able to discuss about it. Finally, the Scrum Master in their team isn’t a pure Scrum Master, just doing tasks assigned to Scrum Masters; he is doing some coding tasks and training.

### 4.2.6 The Sixth Interview

The sixth interviewee, an employee in Volvo IT in Gothenburg, Sweden, has degreed in Computer System and Computer Science. He became an external consultant in Volvo in 2010 and got employed at Volvo IT in 2012. He is now working with a distributed agile project, employed as an application software architect for PBS project. He is also the team leader and Scrum Master for the Sweden team. He has another role acting as Scrum of Scrum Master for the whole team.

In the interview, he mentioned that based on their setup and what he heard from other projects, it’s common that Scrum Masters have other responsibilities like being an
architect or do some developing tasks when it’s needed. He explained that in their setup, testers are working in a separate team called test center, located in Poland, and developers, architects, Scrum Masters, etc. are working in development center in three different locations. In their previous setup, all of them were working in a same team as code center but after increasing in the size of members and with “bad influence of some inexperienced project managers”, they decided to be more “waterfallish” so development center delivers to test center and test center delivers to the customer.

The interviewee claims that “we’re agile in some ways but in some ways we’re not agile”. He continues “our business side thinks that we’re agile because we have two production releases per year and we have a PBL with items that builds up the release. This is something that they think that we’re agile because we’re not doing a lot of items and then having a release after ten years. Also some of these items for one release can be 600 hours that is a project in itself. They’re also not acceptance testing the items until they get the release so even if the item has been done and delivered, a couple of months before the release, they will not acceptance test it and they don’t acceptance test it with end users. Because of that, it’s not possible for us to change the way that end user wants it until next release” so he believes that they aren’t agile with the business side. The releases is the only chance for end user to see the product and for release, they release to one customer and this customer releases to several customers that they are supporting for the project.

He describes that around two and three years ago, the travel budget for their project was high and teams from Poland were traveling to Sweden every third week for one week. The goal was for both Release Planning and Retrospective meetings and also for teams working together and having the feeling they’re in one team. Now, he says that they don’t travel too much since the travel budget is decreased. For knowledge sharing, there is quiet limited amount of budget but for release, they have couple of developers who travel to Poland to work on Backlog items. For traveling, customer needs to pay it so they need to motivate them to get the money.

He believes that the main reasons for distributing projects are cost reduction and precision. Another reason is that Volvo wants to be close to customer and there are places like Brazil and Poland that big Volvo factories are located so development teams are assigned there. Regarding cost, he says that if we look at economic calculation, it’s cost effective but for the overall cost, it costs more than non-distributed projects. To support this claim, he continues that there are some feelings and comments from members of their project that “we produce now as much as we were doing two years ago, when we were half the size” so evaluation is needed to check if distributed agile is appropriate for a particular project.

He experienced some problems with time-zone differences between Sweden and Brazil, which is five hours in European Summer. The main problems were because they need to push up all meetings to the afternoon and it usually takes about one day or 24 hours for mails and questions until they answer or receive an answer. If there is a follow-up question, it will take two days so lead times increase a lot. Another problem is that the most experienced development manager is located in Brazil and for installation; it usually takes around half or one day to look at the problem. According to the interviewee, now there is a backup for him but the problem still exists. He believes that there is no struggle to mitigate the problems of time-zone
differences regarding practices of Volvo; just one analysts, on her sense of responsibility for the project, wakes up early to answer questions.

Regarding cultural and work style differences, he believes that there haven’t been too many differences between Sweden and Poland but for Brazil, some problems happened, especially between Poland and Brazil. This problem forced that line manager and chief cultural manager of Volvo travelled to Poland for one week and set teams together to have workshops about how to have better communication and to understand each other culture.

For issue tracking and collaboration tools, they use both virtual Kanban Board and TFS for their project. He believes that using both of these tools bring frustration to engineers when they want to update daily estimation on the board. Currently, they are “putting a lot of effort into having TFS 2012 with Kanban template” which makes things easier for developers and testers. They use Kanban tools because as this interviewee mentioned they run more about Kanban method with compliments of Scrum. To support it, he claims that “We have deployment of resources and any developer is just allowed to work on 2-3 items at the same time. Also, we’re virtualizing our works with the Kanban and tasks break down and we pull the items instead of pushing them. The Scrum part is iteration planning that is one week basically, it’s just the matter of assigning and planning what developers should be part of, what case team and the Retrospective.”

They use videoconferencing for some Retrospective and for some Sprint Planning meetings but the problem is that rooms are few and they are always occupied and they need to book them several months ahead. Personal webcams is not allowed because of bandwidth problems. He believes that using video gives a complete different working environment to feel that they are working together and to see if someone has a question or someone is hesitant or happy or not.

4.2.7 The Seventh Interview

The interviewee is working in Volvo IT in Wroclaw, Poland and he has been in PBS project for more than seven years. His experiences include working around seven years as a software developer, mostly with RUP and agile methodologies, two and a half years as a system analyst and currently a team leader. In Volvo, he started working as a developer and after five years, then he became system analyst for two and a half years, and finally he has stated to act as a team leader for two months. He has a Masters of Science in information technology, graduated from Opole University of Technology in Poland.

He describes the situation and problems of his current project. Five years ago, they started to work distributed between Sweden and Poland and the project was quiet implementation of Scrum as teams from Poland were travelling to Sweden, every third week for one week for Sprint Planning and Retrospective meetings. He believes that planning was easier and beside it, the team integration was really important as they found out later. Three years ago, the third team from Brazil was added and after that just some members like architects and system analysts are traveling to Sweden seven times in year, each time two weeks. For the first days after adding the third
team, they came to Gothenburg and it has been the only time that the whole team met. After the third team joined the project, the problems of time-zone and cultural differences were started for their project. According to the interviewee, the Brazilians, from the start, were trying to be noticed as good software developers and they were using competition tricks for it. They were also added in the middle of the project, where the team was releasing a new version of the software. They didn’t have enough business knowledge and didn’t have any experience either so one mentor per two developers in Brazil was assigned and they could interrupt anytime they had a problem or a question.

Another problem was time-zone differences, which were five hours for Europe Summer Time but with different working hours between sites, the overlapped time between Sweden and Brazil was two hours and between Poland and Brazil was one hour. When Brazilians had a problem they sent a mail and then waited all day for Swedish and Polish members to be able to answer it. After that, European site members waited till 2’o clock for Brazilians to start their work, reading mails, trying the solutions and if they didn’t understand or have new problems, asking for more help. They need to do all that in just one hour. After that, they started knowledge transfer program that one experienced developer from Sweden and Poland were setting in Brazil, just to solve their problems. The interviewee believes that it wasn’t optimum because one member should handle six members in Brazil when he wasn’t able to produce too much work. They were also tried code review practice that each member should ask another member to review his/her code before checking and delivering the code, which were helpful identifying the problems very early.

As he mentioned, there were some other cultural problems. First, Brazilians were eager to have small talk before discussing issues but Poles avoid it and it was offensive for them. Second, a Polish member in Brazil gave direct feedback to members in Brazil in a room where all members were existed, which wasn’t acceptable for them. Finally, a discussion between a Polish and a Brazilian member about a problem caused Brazilian to send mail that Poles aren’t helping them. It forced Volvo IT to have some cultural and team building session between members in Brazil and Poland. The interviewee believes that it was really helpful and shows the importance of team building practices. He also asserts that it’s valuable for teams to have some integrations, meet face-to-face and do things that aren’t related to the work for the purpose of knowing each other more. He claims that not too many people take traveling and team building practices seriously. By using these practices, communication and collaboration is better and people care more about someone who has met before; care about if he/she is good or not at work, if he is happy, etc.

The interviewee believes that adding the third team from Brazil was a really bad idea from the development team perspective but the main factors were cost and also managers’ plan to assign all the maintenance and support for the project to that team. He claims adding this team that aren’t supported by experienced people, who know the system, caused them to be dependent to the other teams for checking everything. They introduced a lot of bugs and the time needed for completing tasks increased too much, around three times for their project. The lead-time is also increased and you need to invest a lot of time and money in the beginning to transfer the knowledge. He also discusses that because Scrum Masters aren’t fully acting as Scrum Masters and
they are 50 percent developers, they aren’t able to support the team to solve their issues.

Finally, he describes that when they were running one project, they were using Scrum. For that structure, using scrum was good as they estimated tasks on a daily basis and they had different kinds of meeting. After adding some projects to their team in one portfolio, they switched to Kanban. According to him, the reason was because they’re running six project in one portfolio while they receive change requests from different projects at the same time so “in this case, Scrum doesn’t work so we made it a bit hybrid”, a mix of Scrum and Kanban.

4.3 Distributed Locations in Volvo

The software developments in Volvo are done in a division of Volvo called Volvo IT. Volvo IT started distributed projects around seven years ago (in 2006) and started combining working distributed and agile around five years ago (in 2008). Before it, the one and only way of handling projects were RUP. Instead of main location in Sweden, the teams are located in various locations around the world like India (Bangalore), China, Brazil (Curitiba), US, France, Poland (Wroclaw), etc.

India: It’s mainly for the development team and is used as a place to reduce the development cost as well as receiving high quality development from skilled people located there. For example, for ALPHA project, 25 developers with 4 Scrum Masters, which are divided into 4 Scrum teams, are setting in India.

Poland: When it comes to checking the quality and testing the product, Poland is the main target. Test teams are located there. Some development teams are also located there.

Sweden: The most experienced people are located in Sweden since they have been involved with projects for a long time so it’s the place that business knowledge is available. The business analyst, project manager, configuration manager and configuration and deployment manager, are some specific roles setting in Sweden. Integration specialists are also located there. Many architects are in Sweden however now architects are located in other places like India, Poland, and Brazil. Just like other places, some development teams are setting there.

Brazil: Brazil is considered as a low-cost center for development, called ‘competitive sourcing’ in Volvo. For the PBS project, the reason for adding the team in Brazil was to assign all the maintenance and support for the project to this team.

4.4 Analyzed Projects

Two main Volvo active projects were analyzed in this research. Due to confidentiality reasons, the names that are used for this report aren’t actual names of the projects.
4.4.1 PBS

PBS is actually one of the biggest projects in Volvo IT running right now, it’s a production system that is used around the world in different manufacturing centers. This project is handling up to six different projects (All from companies which are part of Volvo group) in one portfolio that are responding to the changes demanded from the customer. The whole team is a continuous release organization and it’s a maintenance team doing new features. There is just one shared Product Backlog that six projects adding and prioritizing the items in it.

PBS consists of three teams in Sweden, Poland and India. The whole team was working as two teams in Poland and Sweden for four and half years but three years ago, the third team from Brazil was added to the team. The geographical distribution of teams in different locations is that there are developers and architect in Gothenburg, Sweden, developers and architect and system analysts in Wroclaw, Poland and developers and system analyst in Curitiba, Brazil. All of those are parts of one team that in Volvo is called Development Center. Testers are located in Poland and they belong to the team called Test Center. On each site, there is a Scrum Master and for the testers, there is a test manager that acts like Scrum Master of test team. The test center has a test manager that manages several test centers.

4.4.2 ALPHA

The project was started in 2011, August and at the time I was interviewing in 2013, January, they were at the end of the project cycle so the length of the project was around one and half year. The goal of this project was improving an existing system and there were certain things that the team could consider from the existing system but they were more eager to start a fresh requirement and a completely fresh database structure.

It’s a distributed team between India, Poland, and Sweden where development team was located in India, the business team was in Sweden and there was just one architect in Poland. Also, the team has one architect in Sweden and one architect in India.

4.5 Teams and Roles and Stakeholders in Volvo

In each agile project, two general groups are involved: business people (known as business team), and development team. The business team is responsible for higher-level tasks and works with the customer for providing the requirement but development team does the lower-level and programming jobs. Below, different teams and roles in Volvo distributed agile projects are categorized.

Business Team
To understand the roles of different stakeholders from the business side, the procedure of gathering requirements and producing Backlog is explained. The Backlog is created by Product Owner with the help of customer but when an item is ready to start, it’s just a small headline with a small description and the work with the
specification and requirements are done by a cross-functional team. This cross-functional team consists of a **business analysts, a solution specialist** from business and a **system analyst**, an architect, a lead developer and a lead tester and this team starts to look at the problem to solve. Each member considers the requirement and issues from his/her own perspective. This feeds the **business analyst** and the **solution specialist** to write the specifications. They do it iteratively and after that, they have a wrap up meeting to approve the specification. When the requirement is approved, the development team and the Scrum Master analyze them to pick the items for a particular Sprint and based on that, there are discussions with the **requirement analyzing team**. Many of the times, they get back to **customers** on certain points and after that it is assigned to the development team. Business analysts are not always located in Sweden since Volvo is working on different systems, which are located in different locations however the main work for integration of the systems is done in Sweden.

**Development team**
The development teams or development Center, an informal word used for development teams in Volvo IT, consist of developers, architects and system analysts. Usually each development team has most of these roles as for PBS team, there are developers and architect in Gothenburg, developers and architect and system analysts in Poland and developers, architect and system analysts in Brazil.

**Test Team**
For some teams like ALPHA project, each team has a role as a tester but for some projects in Volvo like PBS, testers are working in a team called test team, which is part of Test Center. They are usually located in Poland, working separately from development teams. Like each development team, which has one Scrum Master, test team has a test manager who acts like Scrum Masters of testers. There is also a main test manager, who manages several test centers.

**Architect**
In Volvo, each distributed location usually has one architect. For example, for the ALPHA project there are three architects setting in Poland, Sweden and India. There is also a chief architect, which isn’t a distinctive role and is selected from one of available architects. Chief architect knows the complete solution from the architectural perspective and every developer is expected to show him change in the related to architectural issues. Also, change in the contact between the systems should be reported to chief architect as there are several systems and there is a communication between them. Because at Volvo IT, an application software architect is also expected to be a team leader, architects also usually act as a leader.

**Team leader**
Team leader is responsible for the whole team, all developers, analysts, architects, etc., to transmit and receive information. Team leader is required to report this information upwards and downwards; to the management and also lower layers of members. In Volvo IT, an architect is usually assigned to be also the team leader since it is expected that he/she has these kinds of qualities. A team leader gives support to Scrum Master.
Coach
The role of the coach is to guide the team to do the project in an agile way. For some projects, there is one coach for each team but usually in practice, there is one centralized coach for all teams in one location so one coach can guide several teams in one location while being in contact with the Scrum Master for a better guidance. Sometimes, for a distributed team, a coach goes from Sweden to distributed locations for a temporary time. Scrum Master is the central point of contact for the agile coach.

Consultants
It’s a role similar to coach and has been used especially for some early projects. Consultants were hired to guide distributed teams usually because the speed of guiding them to work on the right track and also working on agile methodology were important. Some teams have permanent consultants and some others use it for a temporary time, which could be from onshore or offshore locations. They are some of more experienced Volvo IT employees. According to the third interviewee, teams could have consultants who act as project managers and Scrum Masters.

Scrum Master
The main duty of Scrum Master, which are selected from people or members who are more inclined to agile, is to make sure that the team is working smoothly, moving on the right track and following agile practices. Scrum Master is also the central point of contact for agile coach. He should always be in contact with the team to recognize the performance problem very soon, and to ease the communication between the team and listen to different ideas. Each team in Volvo has one Scrum Master and there is one Scrum of Scrum Master which are usually selected from one of current Scrum Masters and isn’t a distinctive role.

Customer
Focus of Volvo IT was primarily serving the Volvo group and Volvo Cars so they are the primary focus and customers of Volvo IT. Volvo Trucks is the customer of ALPHA project. Customers are located in different locations like Australia, Sweden, and Ukraine for this project. Also, there are some truck companies like Mack and Renault that are subsidiaries of Volvo Group so Volvo IT is supporting some of their projects. For example, PBS team is working with these truck manufactures. Volvo IT sometimes works with other companies but the reason for it is because of the network and contacts or there is a certain component of another company that Volvo is working with and Volvo IT is able to build it.

Working in Sprint
According to the third interviewee, Volvo IT principle is having 3-4 weeks Sprint. Sometimes an exception, the length of Sprint for the PBS project was one week Sprint for the system test period, because as the seventh interviewee claimed, “it gave them better control over bugs, how many are fixed, how many left, are they close to product acceptance plan values”.

[53]
4.6 Scrum Team Structures in Volvo

As discussed in section 3.4.2.1.3, Sutherland, et al. (2008) brings three different distributed Scrum team structures as isolated Scrum, distributed Scrum of Scrums and fully distributed teams. In Volvo IT, structure of the teams is similar to distributed Scrum of Scrum teams. Teams are cross-functional and have most of required skills except that for some teams like PBS, they have isolated test teams which work separately from other teams. Also, those cross-functional teams are not dispersed across different location and the level of dependency is tried to be low while different locations are usually working on same issues. It’s important for teams to communicate and increase the coherency of the teams. Teams usually have their own meeting in their locations and the connection between teams is possible by Scrum-of-Scrum meetings however during the overlapped working hours, teams try to communicate on the similar works and problems. Teams also have some meetings like Retrospective meeting that all members join it.

4.7 Tools

Different tools are used in Volvo to have a better collaboration. These tools, which are used for communication and collaboration, are elaborated below.

Video
Projects in Volvo are using video as a tool for communication but it doesn’t happen very frequently and is just allowed in conference rooms for the meetings. Using personal webcams for one to one communication is not allowed and it happens rarely. For example, for one of the first Volvo IT distributed agile projects, Volvo gave permission to use personal webcams because time to market was really important for that particular project, as the first interviewee declared. The main reason for not using it is the bandwidth limitation. Using video and videoconference happens in conference rooms located in Volvo IT buildings. It’s equipped with large screen LCD or LED TV while projector camera is used to share videos.

Communication Tools
Different tools are used in Volvo for communication. Microsoft Lync (Previously named as Microsoft Communicator) is used for instant communication and Microsoft Live Meeting is used for teams’ meetings. Live Meeting, an extension of Microsoft Collaboration, offers more features like sharing documents and video however video is not mostly allowed because of bandwidth limitation. IP Phone is another tool used for Volvo IT projects.

E-mail is another tool for communication. In Volvo, according to interviewees 5, 6, and 7, E-mail is usually used for status reporting, setting up the time of meetings, and also for sharing problems and issues. The seventh interviewee claims that the result of meetings, especially Scrum and Scrum meetings, is shared by mail.

Development and Issue Tracker Tools
For assigning the use cases or tasks and updating and seeing the progress of them, different tools are used based on the project. For the Microsoft based projects, TFS is
used which is embedded within Microsoft .Net development tools and for Java Projects, Jira tool is mainly used. Currently, some of development projects in Volvo IT, mainly maintenance projects, are using Kanban method so for better collaboration, different Kanban tools are using and evaluating. The PBS project is using Virtual Kanban for this respect.

**TeamPlace**

According to Volvo website, TeamPlace is a team website and a solution provided by Volvo IT to facilitate information sharing, document management and collaboration. Availability is assured and the tool makes it possible to collaborate regardless of time and place. Discussion forums, announcements, tasks and contact lists are shared in one secured part of the web and different level of authorization from open to secure access is possible. All the details about Sprints with their documents are uploaded in TeamPlace for later accessing so if a new person joins the team, all the use cases are documented there (TeamPlace, n.d.).

**Wiki**

Wiki is a tool for documentation and sharing information. It could be used to share information and update the problems. Wiki is a part of TeamPlace but for some projects it’s not used very often because it has the problem that the tool isn’t supported well by TeamPlace. For the PBS project, according to the sixth interviewee, just Brazilians are using wiki but they are using their own wiki, which are used just within their own team. According to Daniel, this interviewee claims that they are seeing a need for a wiki for better collaboration within the whole team.

### 4.8 Empirical Study Results

This section is provided to discuss about how distributed agile projects in Volvo IT is corresponding to the first and second research questions. This section just offers a brief discussion and summary of interviewees’ experiences about these concerns while more elaborations are provided in chapter 5 while comparing them with theoretical findings.

1. What issues need to be evaluated before starting to use distributed agile software development method?

Some criteria are considered for checking the adaptability and the need of a project to use distributed agile development. These evaluations are related to agile method, distributed development and the combination of them.

The first agile project in Volvo happened when the speed of development and time to market were regarded as the most important factors for a specific project, as described by interviewees 1 and 6. This agile value was an inspiration to start this method and after that, projects have used this method using collocated teams and distributed teams. Customers are another factor that whether they want to collaborate with the project for gathering requirement and receive a release after a specific period. If customers provides a consistent requirement and prefer to receive the work after finishing the whole tasks, investing on agile isn’t rational. In distributed agile
projects, teams and managers receive a delivery of the work after each Sprint but customers receive a release after some Sprints and if they want to change the work, they need to wait until next release (referred to interviewees 4, 5, 6 and 7). Success of agile method depends on high-level technical skilled developers and also members who are well adopted to work as a team, so managers are looking for this kind of team composition when they decide to use agile method, as described by the forth interviewee. Concerning cost, in Volvo projects, end-user doesn't have any representative in the team so there is no cost for it but because of this lack of involvement and because of the period between release, there is a possibility of producing wrong functionality, which adds the total cost (referred to interviewees 3, 4, 6 and 7).

There are some opportunities and challenges in a distributed development. According to interviewees 1, 3, 6 and 7, Volvo as a global company is looking to reach local markets around the world, specifically the places that manufacturing sections of the company are located. They intend to be close to them and understand the needs of the market. Reaching global talents is another motivation to use distributed development. For example, an equipped test center to evaluate the quality of the software development is located in Poland with high-skilled members. Also, India is regarded as a place for reaching high-skilled developers. Reducing cost is the next factor that brings some controversies around it. While it’s possible to reduce the cost by outsourcing to regions with cheaper development cost like India and Brazil, distributing has some hidden costs, which needs to be considered. First kind of these costs is the cost of traveling and team building for a distributed project. Another one is the cost of decreasing the level of productivity and quality of the work when teams aren’t able to adopt with the situations of a distributed project (referred to interviewees 1, 6 and 7).

The structure of a distributed team should be designed before starting the project. In Volvo, the structure is similar to partial offloading. At this structure, the customer representatives and the business team including business analysts, project managers, etc., are located in Sweden and have their meeting there. Some teams have development teams in Sweden while for some teams; all development teams are in distributed locations (referred to interviews 3, 4, 5, 6 and 7). This method has some challenges like the need for high level of inter-team communication between business team and development team. According to interviewees 6 and 7, for the ALPHA project that the level of dependency between teams was high and the overlapped time was low, teams had problems for collaboration so productivity of the whole team decreased. For this project, a team was added in the middle of the project and they didn’t have enough knowledge about it so other teams were required to guide them, which were considered as an obstacle to the pace of development. Distributing roles between sites is also important but for mentioned project, the critical role of chief architect were assigned to a new member of the newly added team which caused problems for both teams and that member.
2. What are the main challenges while using distributed agile method and which practices are needed for coping with these challenges and problems?

Like the theoretical study section, four main challenges of distributed agile method are regarded and the practices that are used in Volvo for these challenges are explained. These dominant challenges are, Scrum and team composition challenges, Time-zone differences problems, communication and collaboration and finally team building challenges.

Teams in Volvo aren’t divided based on architectural layers to just handle a specific part of the system. According to interviewees 1 and 7, the struggle is to involve members from different teams and locations on same tasks so they will be able to cooperate. Teams are also cross-functional and each location has all needed skills to develop and release the product. One exception is about testers that some teams just use external testers who are working in Test Center located in Poland, as discussed in interviews 3, 5, 6 and 7. For these teams, development teams deliver the work to the test center and they deliver it to customer. In Volvo, the number of teams, the composition of its members and the number of members working in one team are changing based on the need for the next Sprint, as discussed in interviews 1, 3 and 6. In addition, the role of Scrum Master is important and he is supposed to guide the team to follow agile method for development and assures that collaboration and communication between members is satisfactory. Scrum Master is a part of the team and not supposed to direct the team as this wrong expectation was the reason for some projects to fail, according to interviewees 1 and 4. Also, for many projects, Scrum Master has some other roles in team acting as a developer or an architect, according to interviewees 3, 5, 6 and 7. These mixed responsibilities cause problems in the critical phases of project like before a release. In addition, there is a role as Scrum-of-Scrum Master, who handles the negotiation and collaboration between scrum masters, however it’s not a distinctive role. The need for a distinctive Scrum-of-Scrum Master for bigger projects is seen in Volvo, according to interviewee 6 and 7. It’s the same for the architect as one of the architects is the chief architect and has a broad architectural view about the product. Finally, Scrum Masters in some projects are looking for a catalyst among team members to incite members to start the work immediately in a good way. This member has a similar role like a team coach and isn’t a distinctive, as discussed by the first interviewee.

Scrum meetings are considered to facilitate collaboration and transferring knowledge between members. In Volvo, some meetings of a distributed agile project are distributed and some are collocated. Daily meeting is an important meeting for teams. For most projects, each team usually has its own Daily meeting locally at the start of the day, according to interviewees 1, 3, 4, 5 and 6. The problems with this meeting is that members just receive information about their teams and for having a broader view, they need to check with other teams to avoid interfering with their work and probable merging problems, as discussed by seventh interviewee. After this site-specific meeting, Scrum Masters of teams, the team leader and test leader (for some teams) have a distributed meeting. After finishing this meeting, they send the results of the meeting by E-mail to all members, explained by interviewees 6 and 7. There are two other Daily meetings in Volvo; one that Volvo project managers in offsite and
onsite locations discuss about their related issues and another one that customer representative has a meeting with main project manager and business analyst, according to interviewees 3 and 5. Sprint Planning is another meeting reserved for planning the upcoming Sprint. It’s a distributed meeting that the whole team joins it when team leader provides a list of new items, architect/analyst explains what the items are about and developers choose items and estimate the time they need for implementation, according to interviewee 5 and 7. Retrospective meeting is another Scrum meeting that the positive and negative points of last Sprint are discussed. It’s a distributed meeting but some teams in Volvo just use audio for it, not video, because of bandwidth problems. Retrospective meeting is also used for team building practices, as discussed by interviewee 3, 5, 6 and 7. According to interviewees 5 and 6, Release Planning is another meeting planned for each Sprint that release manager, before any release, handles the release requesting resources planned for the release with some other managers to sync the releases with dependent systems. Finally, after ending each Sprint, a demo is released internally within the team targeting test, requirement and architecture teams, as discussed by interviewees 1 and 4.

Volvo distributed teams are located all around the world in different time zones so various time-zone differences occur while they are working together. Some main practices to solve this issue are ignored in Volvo for low time-zone differences. First is that there is no struggle to increase the overlapped time, as captured from interviews 3, 5, 6 and 7, while there are teams working between Brazil and Poland with one hour overlapped time and have problems for collaboration with their limited overlapped time. Also, the practice of using some members to work early or late to answer the problems of other teams is ignored. As another practice to solve the issues of time-zone differences, some teams, like development teams located in India, have tried to decrease the dependency between distributed teams, (referred to interviewees 3 and 5). In contrast, there are teams in Brazil, which are dependent to other teams to solve problems and to be guided but they need to wait half a day or a full day to receive a reply from other teams so the lead time increases for them, as discussed by interviewees 6 and 7. The distribution of members for that project was another issue as one main role, the deployment manager, was setting in Brazil, which caused problems for other teams. A practice used in Volvo to solve this issue was assigning a balanced role in Sweden. For high time-zone differences, Volvo shifts the working hours of some teams and use a proxy as onsite coordinator to arrange things required for offshore teams, according to the forth interviewee.

Communication is the next main challenge in Volvo distributed agile projects. To have more communication within the team, it’s required to encourage members to communicate more while communication tools and media are provided and improved, as discussed by interviewee 1 and 3. Members working in different locations tend to work in silence. In Volvo, it’s the duty of scrum master to connect members for possible problems, encourage them to fix problems together, and in general creating a communicative environment. According to the first interviewee, one practice to create this environment is pair programming and dividing the responsibility of tasks between sites. Another practice used in Volvo is code reviewing that every member in the team is required to ask another member to look at his/her code and provides feedback about it, as discussed by seventh interviewee. These practices have both social benefit of increasing communication and coherency of team and technical advantage of improving the quality of the work. Different communication tools are used in Volvo.
Video is a tool provided in conference rooms located in Volvo but it’s not a common tool for all teams mainly because of bandwidth limitation and the low amount of available conference rooms in some locations (referred to interviewees 3, 4, 5, 6 and 7). Some teams use this tool as a way to build teams by having online social gatherings and some use it for Retrospective and Sprint Planning meeting. As discussed by the third interviewee, E-mail is another tool, which is usually used for status reporting, sharing problems and issues and sharing the results of meeting. Also, different issue tracker tools are used in Volvo but these tools are regarded as time-consuming tools and the need to simplify them is seen for some projects as some teams use two issue tracker tools at the same time. According to interviewees 6 and 7. In addition to virtual board, some teams have physical board in their site.

Considering the relationship with customer and the triangle of scope, time and budget for a project, Volvo agile projects are flexible on scope and trading on time is possible up to a certain point but the budget and money is mostly fixed, according to the first interviewee. Also, the accepted change is change on the developed product, which comes from feedback of customers and users. As discussed for the first question, after ending each Sprint, there is a delivery within the team but customers receives a release after some Sprints which usually takes around six months between two releases and it’s the only chance for customer and end-user to give feedback. After giving feedback, the customer needs to wait till another release for changes to be effected (referred to interviews 3, 4, 6 and 7). The third interviewee believes that it’s because customer expects to have a complete patch of functionality after some months and the sixth interviewee confirms it and claims that it’s a coaching problem.

The last main challenge is related to team building. In a distributed environment, members with different cultures are working together and they bring their own culture into the work. It’s important to notice that these differences exist and needs to be considered before it cause problems to daily working routine, as confirmed by interviewees 1, 6 and 7. A serious cultural and work style problems happened in PBS project between Poland and Brazil that stopped them from doing their daily tasks. It mainly occurred because they didn’t know about the culture of another site and it was ignored at the start of the project. After doing some team building practices, teams were able to work smoothly, as described by interviewees 6 and 7. The six interviewee claim that it’s like a slogan in Volvo that it’s required to know work people you’re working with however it’s usually ignored. The main focus in Volvo for team building is for the long and critical projects or projects which are working for a long time but aren’t able to match well (referred to interviewees 1 and 7). The budget of the project is now a factor that team building is ignored for some Volvo projects, even for critical projects and projects which are planned to work for a long time, as confirmed in interviewees 3, 4, 5, 6, and 7. For the ALPHA project, there was no team building practice at the start of the project. For PBS project, teams met at the first week of joining the Brazilian team but there was no other gathering until problems occurred for the team communication and collaboration.

In addition to the seeding visits at the start of the project, teams in Volvo usually have some level of maintaining visits in the middle of the project. The second interviewee describes the cycling plan practice that for a project, developers from both offshore and onshore locations were travelling to other locations. Now, for projects in India, developers are not travelling anymore but some experienced members from onsite
location travel to India, described by interviewees 3, 4 and 5. Some practices are used in Volvo to receive the most benefits from maintaining visits. Business knowledge transfer program is organized in PBS project to guide the new team. At this practice, there was always one experienced developer, either from Poland and Sweden, staying in Brazil to train them, answer the questions and introducing the project, according to the seventh interviewee. This practice could be done for other roles to transfer technical or business knowledge. Another practice, described by the first interviewee, is the ambassador practice that people who travel back and forth from different sites, acts as cultural ambassador of that cultures. Rotating team members is another practice required to achieve consistency between sites. On the other hand, the critical parts, recommended to travel are for Sprint Planning meeting, user acceptance testing period, during major release and finally the last iterations to have a smoother release, as explained by interviewees 1, 3, 6 and 7.
5 Analysis and Result

In this section, the results of theoretical section were compared and analyzed with information captured in empirical part of this study to validate and verify the theoretical findings. This section is divided into two sections while the aim of both parts is answering research questions. The first part is a comprehensive analysis of theoretical and empirical data and the second part covers results of this section by summarizing the first section and answering research questions.

5.1 Comprehensive Analysis

5.1.1 Distributed Agile Evaluation

This part of comprehensive analysis is regarded to cover the issues of the first research question.

5.1.1.1 Agile Evaluation

Before starting to use agile for a project, some analysis and assessments are necessary to check if agile is the appropriate way of developing the project.

**Speed:** Speed of development and the need for delivering the project soon is an important driver for using agile as discussed in section 3.4.1.1. The first agile projects in Volvo happened when the time to market were important for them, according to the first interviewee.

**Iterations and Delivery of the software:** Agile gives the opportunity to develop, evaluate and test the project on a daily basis so if it’s required by the customer and the manager of the project to have a constant view about the project, investing on agile is more rational. The number of releases is another factor that whether it’s important for customer to have a release after a specified time. For distributed agile projects in Volvo, the number of releases to managers and to the customer is different. Managers usually receive the product after each Sprint for evaluation and they check it with Scrum Master for improvement but customers don’t receive the work after each Sprint. As discussed in section 3.4.1.1, the first agile principle discusses that customer satisfaction is required through early and continuous delivery of software. In Volvo, as captured in interviews 4, 5, 6, and 7, customers receive a release after some Sprints and even if they need a change on the release, they should wait until next release to the customer for their required changes to be affected. This interaction level with managers and customer shows that in Volvo IT practices, customers are ignored for the delivery while agility is more about managers need to have an evaluation of the work. It also shows that the reason for short iterative development cycle is for managers having a view about the progress of the work. I discuss more about it in section 5.1.2.4.1.

**Requirement:** One of the main drivers for iterative development is to have flexibility on requirements when customers are involved and participated as a team member for
planning and development. As discussed in section 3.4.1.1, agile doesn’t try to follow a pre-specified rigid plan and emphasizes on “good design to improve quality”. When the customer has some business goals but she is not certain about the requirements, agile method is suitable since the requirements are flexible to be changed at any stage. As the forth interviewee mentioned, in situations that requirement is consistent, straightforward and well documented and it’s apparent what needs to be done, what are the sources and the only concern is about the development part, using agile isn’t too much necessary considering business and customer parts since not too much interaction is required with customers. On the other side, if it’s required for the customer to involve in the requirement gathering process, agile is suitable for this purpose. According to interviews 1, 6, and 7, using agile is not just about working on Sprints but requirements should be developed with the help of customers and team members. The capability to change the requirement is required in agile so after gathering requirements very early, the whole members and customer should be able to reread, comment, analyze and give feedback to the business side of the project very early. As discussed in section 3.4.1.1, interaction of business people and developers and their collaboration for the project are parts of agile principles. Finally, as discussed in section 3.4.1.1.1, requirements are not all about development requirements but business requirements should also be considered in terms of budget, schedule, resources, team, and technology.

**Customers:** As discussed above in the requirement section, the question is that what customer really needs and how she wants to interact with the project. It’s necessary to check if customer prefers to give requirements to the team and come back after some months for the result or not, the customer asks to spend some time with team for the requirement and other phases, which for the latter agile works well. I discuss more about involving and collaboration with customer in section 5.1.2.4.1 of this chapter.

**Team Structure:** It’s also important to evaluate the structure of the team and its members. As discussed in section 3.4.1.1.3, the success of agile method depends on high-level technical skilled developers. If one of the team members leaves the group, the project will be affected hugely and the present of an uncommitted team member has a negative effect and may cause to the project failure. Team working is also important in agile so if the team consists of solo performers, the ones who aren’t comfortable working with the team, agile won’t work well for that team. The forth interviewee believes that for agile, ten average members work better than ten professional ones who can’t work with the team. Moreover, team members should be committed to the project and spend significant time in the project.

**Cost:** Agile itself is known as a cost effective way of development since it focuses on producing what customer needs from the start of development however there are some other costs like cost of training and user involvement. In Volvo projects, the end-user doesn’t have any representative in the project and customer representatives and Product Owner receive the product after some Sprints, which usually takes around six months (referred to interviewees 3, 4, 6 and 7), so there isn’t any cost for user involvement but teams might produce wrong functionality in some Sprints, which adds the total cost.
5.1.1.2 Offshore Distribution Evaluation

Some evaluations are needed before distributing an agile project:

**Global Market and Local Access:** As discussed in section 3.4.1.2.3, reaching global market and proximity to local market is one of the main drivers for companies for distributing their projects. According to interviews 1, 3, 6 and 7, a big and global company like Volvo, which has different manufacturing and R&D sections in different countries and are selling their products around the world, should also have access to local markets. It’s important since if a company has a factory in India, they need to be close to them to support from there and understand the local language and culture. Also, it’s necessary to be close to customer and understand its needs. As a result, especially for projects that are targeted various countries and needs to be localized, it’s necessary that these locations also participate in the development. For the Yahoo! distributed case discussed in section 3.4.2.1.4, their plan to localize their software to different regions was their main driver to have international teams in different regions.

**Cost:** Outsourcing just because of the cost has brought many debates. As discussed in section 3.4.1.2.3, companies are trying to reduce the cost by outsourcing to regions with cheaper development cost like India, China, Brazil, etc. and sometimes they could reach this goal but according to the first interview, distributing just because of the cost and not seeing other impacts of distribution is a mistake that cause problems and ineffectiveness; be a reason that projects fail. The first interviewee claims that he considers distributed agile as another way of development the software since it’s a better way of producing software considering reaching global market and talent. As discussed in section 5.1.1.1, considering hidden cost of outsourcing is also important. Cost of traveling and team building practices are one of them. Some teams ignore traveling and team building practices to reduce the cost. By avoiding it, the result is producing low quality software, having communication problems and also decreasing the speed of development. These factors are reasons that total cost of development and delivering the software is increased, as confirmed by interviewees 1, 6 and 7.

**Global Talent:** Reaching global talents and skills is another driver. For example, considering the Volvo development center in Poland, which was first established because of cost issues; then, after seeing the positive results, it’s now considered as a center for high quality software development and testing. Poland is now one of the main centers for testing the projects, according to interviews 1, 6, and 7. Volvo is now trying to expand that test center. India is also another target for reaching high skilled developers.

**Productivity:** Some factors decrease productivity of an agile distributed project. Mainly, it happens when teams are working in different locations and they have issues on cultural differences and have problems with communication and knowledge management but managers aren’t able to use proper practices to consider the issues. According to section 3.4.1.2.3, there is a belief that distributing the work between different time-zones enables 24/7 production as a way to response time to market pressure and to increase productivity. The argument about it is that there is no positive point working 24/7 when the overlapped time is low and teams have problems on
collaboration, communication and integration. These problems are reasons for lower productivity.

5.1.1.3 Team Structure and Distribution challenges

Considering two different levels of distribution, partial and complete offshoring, which are explained by Phalnikar, et al. (2009) and was discussed in section 3.4.1.2.1, the structure in Volvo is similar to partial offloading with slight differences, as captured from interviewees 3, 4, 5, 6 and 7. In the structure, the customer representatives and the business team including business analysts, project managers, etc., are located in Sweden and have their meeting there. For some distributed teams, one or more development teams are also located in Sweden. The situation for architect is different and they are not limited to be in Sweden near business people as suggested in section 3.4.1.2. For both ALPHA and PBS project, architects are located in Sweden and also in all distributed locations involved. For ALPHA project, there are architects locating in Sweden, Poland and India and for PBS project, they are located in Brazil, Sweden and Poland. Among these architects, there is one architect assigned as chief architect that is not necessarily located in Sweden. For PBS project, there have been chief architects in Both Poland and Brazil, according to interviews 6 and 7.

Partial offloading has some challenges. As discussed in sections 3.4.1.2.2, too much inter-team communication is required as business team and development team are located in different locations and as observed in interviews 3, 4, 6 and 7, different levels of communication between these teams are required. The literature suggests that for longer projects, complete offshoring works better but Volvo doesn’t use this structure even for a long project like PBS and they tried to compensate it with high level of traveling at the beginning of the project. For this project, there are development teams located near business team in Sweden so there is also close interaction between development and business team located there, which can mitigate the problem.

The level of distribution is another important challenge. If there are teams in different regions working in various time-zone differences, it’s required to reduce the level of dependency between them so they won’t need to communicate frequently. As an example, for PBS project, which the overlapped time was just one hour and the Brazilians were dependent too much on other teams for guiding, helping them through problems, etc. lots of communication problems occurred that is discussed later in communication and team building discussions in sections 5.1.2.4 and 5.1.2.5. For teams which don’t have enough support in their locations and are working in different time zones, answering questions and solving their problems could take more than 24 hours which makes the work ineffective and is considered as an obstacle for other teams as described by interviewees 6 and 7. At this situation, transferring knowledge and team building practices become important to mitigate the problem. However communication is important for the team coherency as described by the first interviewee, in situations that the overlapped time is really low, low dependent teams is the answer.
The way a distributed team is added to a project and how managers handle and give responsibilities to this new team requires careful attention. According to interviewees 6 and 7, one major problem that happened in one of Volvo projects (PBS) was that a team from Brazil was added to current teams in Sweden and Poland, in the middle of the project, without asking the opinions of members in those two sites. The philosophy behind it was to assign all the maintenance and support for the project, later after finishing the project, to this cheaper Brazilian site. The problem was that the team, which was lacking business and system knowledge, added to a project that was working smoothly. They needed to be guided which were an obstacle for the pace of development while they were introducing bugs to the system. They also interrupted other teams to ask questions, double-check everything, etc. According to the seventh interviewee, when the estimated time for tasks were, for example, 40 hours for developers in Sweden and Poland, for the developers in Brazil which added in the middle of the project and didn’t have any support from experienced members, who can guide them very fast, point them to right direction and solve issues and answer questions, were 120 hours so the proposed time for the whole team were 120 hours. As a result, it takes around three times more to develop tasks which makes it ineffective, time consuming and costly. Another problem was that proper team building practices was missed for the added project and caused cultural problems, which is discussed more in section 5.1.2.5. After two years of adding the Brazilian team, the other teams believed that they were producing as much as they were producing before, when the size of the team were half. As a result, adding this team wasn’t a good idea from development point of view. Another managing mistake for this project was assigning the chief architect role to an inexperienced member from Brazil. As a result, this person quit the project since it was hard for him to handle all of the responsibilities. In general, according to interviewees 1 and 7, assigning a distributed team is just the start; more importantly, you need to support the team. At the start, teams usually don’t have enough business, system and cultural knowledge so both guiding them and building the coherency of the team become important. Adding a distributed team and expecting that they can work smoothly with other teams without team building practice and transferring knowledge is a mistake resulting mentioned problems.

5.1.2 Distributed Agile Challenges and Practices

This section of comprehensive analysis concerns about the issues of the second research question. The main challenges of a distributed agile project are divided into four sections and for each section, the findings of empirical study are confirmed and compared with theoretical study results.

5.1.2.1 Scrum Roles and Team Composition Practices

Some practices and considerations are needed for teams and its members to adjust to Scrum and perform their roles more efficiently in the team. I discuss more about it in this section.

As part of Volvo guidelines for distributed agile projects, teams in Volvo are not divided based on architectural layers and not separated to handle just one part of the
system, according to the first interviewee. The seventh interviewee, part of PBS team, also believes that having two developers handling a server part, three developers handling client part or another developer or two developers handling some database upgrading applications could bring an unhealthy competitive environment. He continues that by using this structure, there will be three independent teams with deepen knowledge of technical skills and business knowledge but on one area and they all want to prove that they are better and more skilled than others. The situation is similar for distributing the work across different locations. Assigning all the responsibilities for a particular part of the project to one place could make distributing practices even worse since members of a site try to pass the problems and make other sites guilty about the bugs and errors. As a result, involving developers from different sites has the benefit that a team does a part of the work and passes it to other teams. If a problem exists or some investigations are needed, other teams could take responsibility for it and send the result of evaluation to the team looking for help.

While distributing the roles for the development teams, it’s suggested that each location or preferably each team has all needed skills including architect, system analyst, testers and developers (Miller, 2008; Pham and Pham, 2011). It’s also suggested that teams be cross-functional however this fundamental rule is ignored for some projects in Volvo particularly because they don’t have any tester working with development team, as discussed in interviews 3, 5, 6 and 7. For these projects, teams rely on test teams in Poland however the Volvo guideline for distributed agile projects recommends that each team has all needed competences including testing, developing, analysis, etc. so they can produce and test the software and they can ship it, as mentioned by the first interviewee. The current practice for some teams is that the developed software is delivered from the development team to the test center and the test center delivers it to the customer. As the first interviewee asserted, Volvo is even trying to expand this center.

Rotating team members is another useful practice. In Volvo, teams are not constant and it keeps changing based on teams’ planning for the next Sprint, as explained in interviews 1, 3 and 6. For creating the teams, analyzes are done based on needed knowledge and skills for the next sprint. The number of teams is also not constant and it depends on different needs and different phases of the projects. In addition, the number of members in a team is also varying however teams are small (between 5-9 people in each team) while based on different phases of the project, it’s sometimes even smaller. This level of rotating happens within the team in a distributed location. I discuss about rotating team members across sites in section 5.1.2.5.

Scrum Master plays an important role in the team but some teams have wrong understanding about this role. Scrum Masters should guide the team to follow agile way of working for development and make sure that collaboration and communication between members are appropriate, as discussed by the first interviewee. Scrum Master also has Daily meeting within the team to report the daily status of the project to Scrum Masters of other teams. He is a part of the team, not a manager and is not supposed to direct the team. The forth interviewee brings an example, not in Volvo, that a project failed because of this wrong expectation and understanding of Scrum rules and because team members weren’t able to map to this.
Regarding the importunateness of role of Scrum Master, It’s suggested that there be a distinctive member handling the tasks of Scrum Master but Scrum Masters in Volvo usually have some other roles in team as they usually do some coding level works or act as architects, as discussed by interviewees 3, 5, 6 and 7. For example, Scrum Master of Swedish team for PBS project is also an application software architect and team leader and he also acts as Scrum of Scrum Masters for the whole team. According to the seventh Interviewee, Scrum Masters in their team acts 50 percent of their time as a developer and 50 percent as a Scrum Master. He believes that it’s not efficient since they need to either focus on their roles as Scrum Master or focus on development. Especially when teams have product delivery and it’s time for fixing bugs and preparing the deliverable product, the Scrum Master is really important but they are supposed to do coding even sometimes more than 50 percent of their time because of some critical major bugs of that period.

In addition to have Scrum Master and architect for each team, each project should have someone to take responsibility for the whole Scrum teams as Scrum of Scrum Master and a member to act as a chief architect. Regarding Scrum Master, by increasing the number of teams and the level of interaction between them, more collaboration between Scrum Masters is required so a Scrum Master should take responsibility for it. From architectural point of view, one chief architect is needed that every developer in the team expected to show the chief architect, every changes in the most critical parts of the system, any architectural issue or change in the contact between the systems, as discussed by interviewees 6 and 7. These roles are expected for PBS project in Volvo because there are several systems which are communicating so any architectural change must be discussed with the chief architect who knows the complete solution from the architectural perspective.

There are some factors to decide if it’s needed that Scrum of Scrum Master role be distinctive. For the Yahoo! Case explained in section 3.4.2.1.4, each region had one distinctive Scrum Master and there was one core Scrum Master for all regions. In Volvo as discussed by the second interviewee, each team has one local Scrum Master and from all Scrum Masters location in one location, there is one Scrum Master, similar with regional Scrum Master in Yahoo! Case. This regional Scrum Master is for communicating with other regional Scrum Masters locating in other locations. The regional Scrum Masters are usually selected from one of current Scrum Masters and are not distinctive roles, confirmed by interviewees 6 and 7. Also, there is no distinctive role as chief Scrum Master. When the number of teams is increasing, the need for more coordination is seen as in Volvo, for some projects in India or other locations, they have more than four teams working together.

Finally, Finding catalyst for a team is a practice followed by some Scrum Masters in Volvo, which is useful for new teams and teams that aren’t well adjusted to agile way of working. In this practice, Scrum Masters are looking into the team to find one or some members as catalysts to have the role of inciting the team to work immediately in a good way. Catalysts could be a developer, tester, an architect, etc. The first interviewee believes that an active architect could fit well in this role, since they are also leaders in a sense. He must be able to control and survive well in an uncertain environment. According to the first interviewee, it’s important for mangers and Scrum Masters to make catalysts their pairs so by engaging with them and giving them trust and some level of freedom, team members understand that they can lead. He believes
that by using this practice, it’s worthless to add someone to the team as leader since catalysts can do this role. Also, if team works for a while and you add someone from outside to it as a leader, members won’t listen to him/her.

5.1.2.2 Scrum Meetings

Scrum meetings are useful for better collaboration between members, for transferring knowledge and also for discussing problems and issues. The structure of the team and the way members are distributed between locations requires some meetings be distributed and some collocated. The time of a meeting is also important. Sometimes it’s needed that each team has its own meeting but when required that all members join it, a convenient time for all teams should be selected. If it’s not possible, a convenient time for one site is selected and the time is changed for the next weeks to be convenient for teams in other locations. It’s required to avoid exhausting for one team as described in section 3.4.2.1.5. Other challenges for a distributed meeting are less time availability because of being distributed and necessity to have more time for arranging the meeting and discussing the issues. As a result, pre-works for participants are needed before meetings to gain a better result from it. Below, the challenges, practices, and suggestion for common Scrum meeting are described.

Daily Meetings

Stand-up or Daily meeting is considered as a critical and not ignorable meeting for all teams in Volvo. As discussed in section 3.4.2.1.5, this meeting provides a coordination mechanism and is helpful for members to get to know each other and also to reveal problems early on. Some projects have distributed daily meeting between all members while others prefer to have site-specific daily meeting. For the latter, each onsite and offsite team has its own Daily meeting, which is between 15-30 minutes at the start of the day. It’s between the members of development team. In Volvo, for projects that have separated test teams, they don’t join this meeting and they have their own meeting. On the other hand, a particular member of other teams is asked for a localized meeting if there a dependency and she/he is available at that time. Scrum Master or test manager (for the test teams) manages this meeting. It’s a synchronization meeting that members discuss and answer three Scrum questions, “what’s the status, what we should do today, what impediments are in your way?”.

There are some problems and disadvantages with having just localized Daily meetings. Since practically, it’s not possible that, for example, 30 members join in one meeting and all share their ideas in 15-30 minutes, site-specific meeting is used. As a result, members just have the chance to understand what their team members are doing in their location so they need to check with members in other locations to understand who are working in the same area to avoid interfering with their piece of code and prevent probable merging problem later as described by the seventh interviewee. On the other side, when it’s required that all members join the meeting, pre-work before it is helpful to minimize the time needed for the conference and to overcome language barriers. As discussed in section 3.4.2.1.5, members could answer and sent the answers by E-mail so other members could have a better view about the work of members of other teams. For teams with high time-zone differences, using a proxy who participates in meetings of different locations is useful.
The outcome of this meeting is important for the team coherency. According to the first interviewee, the result of this meeting is sometimes just a status report to Scrum Master, which is not acceptable. He recommends that the next level of matureness in Scrum should be talking about “what we should do today” to have a daily plan. Scrum doesn’t regard daily plan but XP recommends it. It’s helpful to have a common theme for the day and for teams to work on the same page. Also, discussed in section 3.4.2.1.5, Pham and Pham (2011) believes that unlike traditional method, the role of this meeting is for teams gathering every day to inspect and checking the team’s progress toward the spiritual goal and not just tracking project status. After this meeting, teams usually during the day have open communication discussing issues and problems. It depends on the structure and openness of the teams that how much ideas are shared in this meeting but if it doesn’t happen, teams rely on Retrospective as another option.

Scrum works really well and adjusted for project managers, as discussed by seventh interviewee, if members estimate their tasks on a daily basis that how many hours are required to finish their tasks. Re-estimating on a daily basis is also necessary since members have new views about tasks every day. The estimation should be reported on a daily basis to the project manager so he/she would have a clear view about the time needed to finish tasks in the Product Backlog. This practice works well for bigger teams.

After stand-up Daily meetings, there are some other Daily meetings that some specific members join it. For teams don’t have distributed Daily meeting, there needs to be a Scrum of Scrum meeting where all Scrum Masters, the team leader and the test leader meet to share the information from the Daily meetings and discuss about impediments, planning and resource location; issues related to team working, as described by interviewee 6 and 7. After the meeting, the results and minutes of the meeting, which actually explain what each member of the team is doing, are sent in an outlook mail to all members. The problem with sending the results with E-mail is that not all members read it in a daily basis however if someone is interested to know if other members aren’t interfering with his work, it’s useful. For the ALPHA project, there is another Scrum-of-Scrum meeting which happens everyday where Volvo project manager, the project manager setting in Sweden, project managers in offsite locations, the business analyst and the test manager join it, according to interviewees 3 and 5. Also, there is another Scrum-of-Scrum meeting, which involves the project manager, the business analyst and the customer representative so Scrum meeting is also going along with the customer as well in a daily routine. Discussed in section 3.4.2.1.5, some teams prefer to have weekly Scrum-of-Scrum meetings but in Volvo, teams usually have these meetings in a daily basis.

**Sprint Planning**

Planning the upcoming Sprint is the goal of this meeting. The fifth interviewee believes that one of the main benefits of agile for developers is reserving enough time for planning and taking up tasks by themselves, so there won’t be any lagging for the project. She continues that for their project, the first week goes for the planning. The whole team including Scrum Masters, system analysts, architects and developers (depends on structure of the team, sometimes just lead developers) joins Sprint Planning meeting which jointly have all needed information for the Sprints. The seventh interviewee explains the role of different members where team leader
provides a list of new items, architect/analyst explains what the items are about and developers estimate the time they need for implementation. The first two mentioned sections, providing and explaining the items, are related to what categorized in section 3.4.2.1.5, as the first part of Sprint Planning meeting; answering “what” question and discussing about the requirements and user stories. The second part goes for answering “how” question, when the development team tries to identify tasks from the previously chosen stories and deduce how much, in hours, it will take for them to turn these tasks into potentially shippable product increments. At the same time of the Sprint, Retrospective meeting takes place to discuss about what was fine and what was wrong during last Sprint that is discussed in the upcoming section.

**Retrospective Meeting**

Retrospective meeting is planned at the end of each Sprint. All members join this meeting and they have the opportunity to discuss about the obstacles and negative and positive points of previous Sprint, as described by interviewees 3, 5, and 7. Based on that, the possible improvements and changes for the next Sprint is planned and if required, the way of working is changed for the next Sprint, discussed in section 3.4.2.1.5. Also, retrospective meeting is useful for risk review and risk evaluation and for finding new risks. It usually takes between 1-3 hours depending on the length of the Sprint and size of the team.

In an open team, developers should be free to discuss about their ideas especially during Daily meeting but when it doesn’t happen, this meeting is a good way for developers to do it. At this meeting, some members aren’t comfortable talking and sharing their opinions so a recommended practice is to pass papers to all members, ask them to write their ideas. After collecting it, someone is required to read all loudly, as mentioned by interviewees 3 and 5. Some parts of Retrospective meeting are focused on a pre-planned topics and areas.

As discussed in section 3.4.2.1.5, Retrospective meeting, like Sprint Planning, is beneficial for improving communication, coordination, and team cohesion in a distributed project. In Volvo, some teams because of bandwidth problems just use audio for this meeting however it’s always recommended to use video for Retrospective meeting because of the goals of the meeting. Retrospective meetings are also important regarding social aspects and team building factors. Teams use this meeting for celebrating virtually for a birthday or another kinds of celebrations. Also, teams that are losing the connection between distributed members are using this meeting as a tool to solve the issue, captured from interviewees 3 and 6.

**Release Planning Meeting**

Before any release, release manager, or another relevant member, handles the release requesting resources planned for the release with couple of other managers to sync the releases with some dependent systems, as discussed by interviewees 5 and 6. However other members like architects, lead developers, lead testers, lead analysts don’t join this meeting, they have a proper understanding about the release since they usually look at ‘time plan’ and ‘PBL’. It’s a part of agile and Scrum principles to have a release after each Sprint but in Volvo, usually after some Sprints, there is a release. Release also depends on the customer, sometimes there might be a bug in the system and customer wants to fix it urgently so teams should have a release however as I
discuss in section 5.1.2.4.2, even for urgent situations, the release is after some Sprints which usually takes around six months.

**Demo Meeting**
After ending each Sprint in Volvo, a demo is released internally within the team mostly for targeting test, requirement and architecture teams. By that, according to the first interviewee, developers and testers can look at the work after each Sprint to give feedback about negative and positive points of it. The forth interviewee claims that aims of this meeting are transferring knowledge and testing the demo. However there is a demo release in Volvo but it’s different with what explained as Sprint Demo meeting in section 3.4.2.1.5 that team members, Scrum Masters and Product Owner join for evaluation of the demo to have a release to the customer and users.

**Sprint Review**
There is no meeting as Sprint Review in Volvo. The discussions anticipated for this meeting like checking what was done and what was not done is usually discussed in Sprint Planning and Retrospective meetings.

**5.1.2.3 Time-zone Differences**

In Volvo, distributed teams are located all around the world, from China, India and Europe to Brazil and United States. These locations are working together for different projects so various time-zone differences from low time-zone differences, like people working from India or Brazil with Europe, to high time-zone differences, like distributed teams working between India and US, occur which needs different practices to mitigate and solve the problems of it. According to section 3.4.2.2, the time-zone differences between three to four hours are acceptable since teams could have meeting and pairing working on the overlapped time but for higher differences, the problems are more and special considerations are required.

The problems of time-zone differences are all related to the fact that you need someone in another place but you couldn’t reach him since he is currently out of work. Synchronization of working hours to increase the overlapped time is useful to increase the availability and to have more communication between teams, as suggested in section 3.4.2.2. One main problem in Volvo IT projects is that there is no struggle to increase the overlapped time for low time-zone difference projects, as captured from interviews 3, 5, 6 and 7. Members are starting and finishing their work like other people in their location so there is no time adjustment for a specific project. The time-zone differences between Europe (Sweden, Poland) and Brazil are varying from three hours in winter time to five hours in summer time. By considering eight hours daily working, when all teams start their work at the same time like 9, the overlapped time is between three to five hours. Interestingly, employees in Poland start their daily work at 7, in Sweden at 8, and in Brazil at 9, which makes the problem even worse, so the overlapped time between Brazil and Poland is just one hour and between Brazil and Sweden is two hours in summer time. That brings the problem that teams need to consider all of the problems and bring the issues in just one or two hours even in a harsh situation that Brazilians were added to the team in the middle of the project and they needed to be guided frequently by teams in Poland and Sweden. According to interviewees 6 and 7, it was usually impossible to consider
all of the problems in this overlapped time so they needed to wait for 24 hours to receive an answer for the issues and problems. By shifting the working hours of Brazil one or two hours earlier and Poland and Sweden one or two hours later, it’s possible to increase the overlapped time. Same practice could happen for the teams working between India and Europe.

Decreasing the dependency between teams is another useful practice to come up with time-zone difference problems. It doesn’t mean that teams need to work in isolation as function-based teams since a function-based team brings the problems of team coherency and communication. According to interviewees 3, 4 and 5, teams in India mostly declared that they don’t have too much problems working with teams in Europe because of the lower dependency and because all development teams are located in India. The second interviewee, working in ALPHA project in India, explains that the style of work is that they start their work, if they have any dependency to other teams, they send an email asking for a meeting. Then they focus on issues not related to teams in other locations. After that, they work on needed tasks together on the overlapped time and finally, Swedes do their work so “it’s a chain of responsibility handing over to the next person and then leaving.”

Teams should be cautious about the way members are distributed between sites to decrease the dependency and increase productivity. In Volvo, according to interviewees 3 and 5, for projects between India and Europe, some of business analysts, some architects and also integration specialists are located in Sweden. When a Scrum-of-Scrum meeting is required, all of them are available for the meeting so more functional and business representatives are located in Sweden. For these teams, Scrum teams in India are more development based and test is also based in India or Poland while architects are setting in Europe. Because of this lower dependency, teams have less problems working distributed. On the contrast, according to interviewees 6 and 7, the dependency for PBS project between teams in Brazil and teams in Sweden and Poland were high in a situation that Brazilians were inexperienced and needed to be trained by Swedes and Poles. The sixth interviewee claims that as another problem for the mentioned project regarding distribution, the most experienced deployment manager was located in Brazil. As a result, if people in Poland and Sweden had a problem with installation or another related issue, they needed to wait for half of a day to one day until that manager can start to look into the problem and sometimes it was problematic since the lead-time increases a lot. This problem can be solved by using the practice of employing someone with the same responsibility in other places as they mitigate it by employing a deployment manager in Sweden to back up the one in Brazil and take care of the problems that needed to be fixed fast however the seventh interviewee believes that problem is still available from time-to-time.

For the dependent team and even teams with lower dependency, the practice of having some members to work early or late to answer the problems of other teams is useful. The member should have enough knowledge about the work of his/her team. If it’s possible, different members should take this responsibility and rotate it between themselves. According to interviewees 6 and 7, this practice usually isn’t followed in Volvo IT for lower time-zone difference teams. For the PBS project, one Brazilian member wakes up early for following issues and answering questions however it isn’t
something that was encouraged by Volvo but it happens on the member’s sense of responsibility for the project.

For teams working with high-time differences that by considering normal working hours, the overlapped time is near to zero, there is no other way to shift the working hours to have at least small overlapped time, for example, moving the working hours to evening for distributed teams in India working with US teams. Another way is using proxy. Discussed in section 3.4.2.2, one kind of proxy is using a representative in offshore team to be active in offshore meeting and pass the results of the meetings to the onshore team. The representative should have worked with the onshore team and have a good collaboration and rapport with them. Finally, a practice used in Volvo is assigning some members in onsite location as bridge gap or onsite coordinator to arrange things, as discussed by the forth interviewee. This practice has the problems of using proxy discussed in sections 3.4.2.1.2 and 3.4.2.1.5.

5.1.2.4 Communication and collaboration

As discussed in section 3.4.2.3, communication is an important part of software development, which enables the team to build a higher quality product. Poor communication has a negative impact on the quality, value and accuracy of the end result. Teams prefer to be collocated to use the feature of proximity for building personal relationship, for better collaboration and also for solving problems together.

There are some communication issues and some fundamental differences between the features of agile and distributed developments, which require attention while using distributed agile method. First, face-to-face communication, which is suggested in agile and is useful to understand the feeling, facial expression, body language, and intention of the one you’re talking with, can’t be achieved in a distributed setting. Another issue, which also discussed in section 3.4.2.3, is that casual conversation and intensive person-to-person communication, which is emphasized in agile, is not possible and sometimes it’s replaced with formal mechanism of communication. As a result, the struggle should be to find a way to have more communication through available channels of communication. According to Shrivastava and Date (2010) and Ramesh, et al. (2006), it requires commitment from team members and also using available tools that we discuss about both of them in the next paragraphs (also referred to interviewees 1 and 3).

One problem is that when teams and their members are in different places or departments, they tend to work in silence thinking about their own part rather than the big picture of the project (this problem also happens in some collocated teams), as discussed by interviewees 1 and 3. The challenge should be to force them to work together as one team. It’s usually the duty of Scrum Master to make sure it happens so he should encourage members to communicate more. According to interviewees 1, 3, 6 and 7, Scrum Master should also connect members for the possible problems to solve them together, makes sure that interaction works and creates a communicative environment that members feel safe to discuss about issues. According to the first interviewee, Scrum Masters need to work more on team’s spirit since projects fails or succeeds as the team so if there is any problem, members need to communicate with a
pair to solve, that helps the whole team to reach the objective. There are some technical and social practices to have a better collaboration.

Pair programming is useful for teams to have more communication and collaboration so it’s important not to give the responsibility of one feature to just one site but divide it between distributed locations for promoting collaboration between sites and between members of sites. The first interviewee claims that it has the advantages that members could learn from each other and if one member isn’t available in one location, there are some members in other locations, which could handle the possible problems. That’s useful to make sure teams are cross-site and there is just one team with one goal.

Another practice that was experienced in PBS project is code reviewing. According to the seventh interviewee, every member in that team is required to ask another member to look at his/her code and provide feedback. In addition to social benefits of it that encourages members to collaborate more, it’s useful for ensuring the quality of the work since the problems are identified very early before the code is checked and delivered.

It’s recommended to avoid using proxies, however sometimes there is no other way just to use it. For example, for Daily Scrum meetings, each team has its own meeting and then Scrum Masters of teams, acting as proxy, have another meeting between themselves sharing issues discussed in the first meetings. Also, according to the first interviewee, in some teams, someone in the onsite location is assigned to be the site manager of an offsite location to transmit the requirements needs to be built to distributed members. In addition, as discussed in time-zone differences section, teams with high time-zone differences also use proxies by utilizing representatives in offshore locations or assigning an onsite coordinator. Using proxy leads to low bandwidth in communication and teams get dependent on few individuals so if they are sick or unavailable, the projects’ progress and collaboration between them fall into problems. Information also gets misinterpreted; if an intermediary is used to transmit messages or problems, he might understand it in another way or some information might be lost along the way, discussed by the first interviewee.

5.1.2.4.1 Communication Tools

Using several different kinds of communication media and multiple communication methods that can be applied in parallel are helpful for mitigating communication problems, as discussed in section 3.4.2.3. These tools are useful for discussing issues and sharing knowledge.

Video

Using video is important to have a better experience of online communication but in general, it’s not common to use video for the distributed meeting in Volvo, according to interviewees 3, 4, 5, 6, and 7. For example, developers, who are working in ALPHA project in India, used video rarely and it only happened once during the period of two years. The situation is even same for Scrum Masters of mentioned project as interviewees 3, 4, and 5 claimed. Interestingly, the forth interviewee, a senior developer, expressed that they don’t “feel handicapped not using it “ and the
fifth interviewee believes “it’s not required using it”. These members met some members from other sites when they traveled to India for the critical parts of the project and never met some of them. The fifth interviewee believes that videoconferencing is not too much required for communication between development teams in India and other teams. To prove his claim, he continued that video is just required for negotiation so because their meeting and communication is about doing the work and talking about the details of it and it’s not about what can be delivered and what can’t be delivered s they have been already planned. Even if we accept that video is just required for negotiation, teams for the Retrospective and Sprint Planning meetings need to have some kinds of negotiation on the work so it’s required to use video for these meetings.

It makes a real difference using video for communication. People are more committed to a person when they know him/her by face. According to the first interviewee, it also offers a complete different working environment to members to have the feeling that they are working together. By using video, members can also see if someone has a question, if he is afraid to interrupt or is hesitant to talk about an issue. In addition, the feeling of participants, facial expressions and body language are partly transmitted, as discussed in section 3.4.2.3. Finally, people have more concentration on discussion when there is a face and it stops them from being multitasking during meeting.

The usage of video is quiet vary. However it’s always recommended to reserve some budget for team building on the first days, videoconferencing can be used as an alternative way for low budget projects for this purpose. According to the third interviewee, video could also be utilized on different occasions to have some kinds of online gathering, for example, to celebrate a birthday, which is useful for building the team’s spirit when members aren’t in one location. Videoconferencing is more recommended for Retrospective and Sprint Planning meeting.

The video facilities should also be equipped. Discussed in section 3.4.2.3, Deemer, et al. (2010) suggests using video for each working room and preferably using it passively. This facility isn’t used in Volvo and with current bandwidth limitation, it’s not possible to be implemented so teams need to use conference rooms. The third interviewee suggests that conference rooms should be equipped by Cisco Tele set using HD TV however it’s a very fancy type of videoconferencing and is quiet expensive. The main problem with these rooms in Volvo is that the number of them is quiet few and for some locations they are most of time occupied and members are required to book them sometimes several months before a meeting (referred to interviewees 6 and 7).

**E-mail**

E-mail is a tool used for asynchronous communication and is useful when one party isn’t available. According to interviewees 5, 6 and 7, E-mail in Volvo is usually used for status reporting, setting up the time of meetings, sharing problems and issues and also sharing the results of meeting, especially Scrum and Scrum meetings. The latter mails are useful to check if members of other teams are not interfering with your work or they aren’t doing the same work but the problem is that not all members frequently check these mails. According to section 3.4.2.3, disadvantages of using E-mail are
wasting more time while communication and losing the mood of the sender however it’s useful for having a record of the discussion.

**Development and Issue Tracker Tools**

According to section 3.4.2.3, development and issue tracker tools are useful for maintaining the documentation, having good transparency and tracking project status. These tools are used to report issues and assign priorities to the tasks. Backlog and Sprint Bundle are shared on these tools and different teams could see and update the progress there. Use cases and documents are added to these tools to plan the work and assign tasks to different members. The problem with the tools is that they need to be updated frequently, which is time consuming.

It’s necessary to simplify development tracker tools and optimize the time spent on them. Currently, the teams which are also using Kanban method, are using both TFS and Virtual Kanban tools so they need to update their daily estimation on both of these tools. According to interviewees 6 and 7, these teams have problems with it since it’s time consuming and is like an interruption for their daily work. The problem is even worse for the Brazilian team in PBS project since the database is located in Sweden while the connection is slow and takes some time for them to use the tool. The sixth interviewee claims that teams are looking for a solution to use TFS with Kanban template so they will use one less tool while the board is integrated with the tool that they are using mostly during the day.

Physical board is another tools that is used in addition to virtual board. According to the first interviewee, it’s feasible to use both task boards like Jira and physical boards. Physical board is used as the master board for meetings while for virtual board is for people who are called from distributed locations to a meeting. The first interviewee believes that assigning tasks are possible by using a strict language like “I’m going to take this ticket with ID and I’m moving that from analysis to development”. One Member could be responsible for updating the TFS while changing the task status on the physical board. He even believes that there is actually no need to have a virtual board; a physical board with a little bit imagination is enough so the board is updated and then a snapshot is taken to send by mail.

For a collocated team, members update the work on whiteboard, the Sprint Board chart, but for a distributed project, virtual tools are used which members need to go and open the system to see the updates. As discussed by the third interviewee, for a better feeling of being in one team, a large screen LCD TV in different locations can be installed so members can sit across locations and when they have standup meetings, they modify the virtual board as they speak. By using this facility, teams see the updates easily and the updates are displayed with an alert and priorities could be dominated. The problem with that is the cost but the results could compensate it.

Finally, Wiki is a tool for documentation and sharing information. It could be used to share information and update the problems globally so less time is wasted, as discussed in section 3.4.2.3.
5.1.2.4.2 Collaboration with customers

Considering the triangle of Scope, Time, Budget for a project, the first interviewee believes that Volvo agile projects like other agile projects are flexible on Scope so while negotiation with customers and Product Owner, they can add or remove some features while Volvo is cautious about what they trade on, not giving all the control to customers. According to the first interviewee, trading on time is possible up to certain point but customers usually don’t want this. Finally, the budget and money is mostly fixed because the only reason that Volvo is trade on scope is the overall cost. In addition, adding more cash to a project requires adding more people so the velocity of the team decreases. The first interviewee also claims that there is a wrong customer expectation that because the project is based on agile, they can change everything and it doesn’t cost anything. He continues that for changes in agile, you need to know the things, which need to be done, and the pace of development. You also need to document them like waterfall so it needs more communication and building and ramping on building. The accepted change is change on the developed product, which comes from feedback of customers and users.

In a traditional agile project, the project is released after each Sprint and customers can see the product and give feedback for the next Sprint but the situation is different in Volvo IT agile projects. In Volvo, at the end of each Sprint, members can see the product internally between the team and the product is released to customer after some Sprints so it usually takes around six months between two releases to customer (The forth interviewee claims that they sometimes have release to customer after two Sprints which is less in compare to other practices). For some projects, the team before the release goes for user acceptance phase with customer that end-user is also not a part of this phase and after finishing it, the product is released to customer. This situation is the reason that releases after some Sprints be the only chance for customer and end-user to see the product and to give feedback. If customer sees any urgent issues or has any suggestion or feedback, the team will consider them for the next release, even for the urgent issues, and as discussed before, the period between two releases is around six months. The sixth interviewee believes that since they are not doing acceptance test with end users, it’s not possible for the team to change the way that end user wants until next release.

Considering the mentioned situation, Volvo IT agile projects are not fully pursuing business side principles of agile method. As discussed in section 3.4.2.1.1, it’s a part of Scrum principles that “the requirement for a new software system will not be completely known until the users have used it”. According to the third interviewee, the release depends more on complete patch of functionality that the customer is looking for and that’s what the customer wants and is the reason to have a late release. The sixth interviewee believes that the end-user and end-customer isn’t aware that they can test it previously while sometimes they don’t want it and it’s more a coaching problem. As a result, when they ask customers to test it earlier, some of them reject it, citing that we don’t want it before it goes into production. He continues that some customers have been started to be interested to do testing before team do the actual release so they have one or two customers taking part in user acceptance test period, a couple of week before the final release, and they’re beginning to take more interest into seeing what’s being done. Frequent deliveries to customer and end
user, which is a part of agile principles, is required to increase the visibility over the project and avoid developing wrong and unnecessary requirements and features.

5.1.2.5 Team building

That’s the nature of every distributed project that people with different cultures are working in the same team and it’s usual that they bring some parts of their culture into their working style and the way they react with their colleagues, setting in different places. As a result, it’s important to understand the fact that these differences exist and need to be considered before they cause problems to daily working routine, communication, collaboration and in general quality of the work.

To have an overview of the situation in Volvo, the cultures of four different countries, Sweden, Poland, India and Brazil, and the way they react with other cultures are reviewed in this research. Indians mostly have no problem working with other cultures and especially they are well adopted and adjusted working with European countries. The time differences are low and they are eager to understand other cultures so no complain was captured about it, both from Indians and Swedes, according to interviewees 1, 2, 3, 4, and 5, however what I received from my interviews is that some of Indians are reluctant talking about problems and shortages. Considering Swedes, however they feel that there are some differences between their culture and culture of other countries, especially non-European countries, it’s a part of their culture to respect other cultures while from work perspective, they treat everyone as equal. Polish people, Poles, have a similar culture compare to Swedes especially regarding working style.

According to interviews 6 and 7, some serious cultural and work style problems occurred between Brazil and Poland. The situation was that, in the middle of PBS project, a distributed agile project between Poland and Sweden, a third team from Brazil was added to the current teams. They were all new to the project and the seventh interviewee believes that as part of their culture, “Brazilians were really enthusiastic and basically they wanted to be noticed from the very beginning as good software developers, skilled ones and they were using the competition stuff, to show off”. That was a cultural difference in compare to Poles. Also, because they were added in the middle of the project, they needed to be trained mostly from business knowledge perspective. It increased the problems since Poles should put time to guide them while they were busy with their work. As another issue, it’s a part of Brazilian culture that before talking about the real issue, they tend to have small talk about things likes weather, Sport, etc. It’s not a similar working style compare to Europeans and when avoided by Poles because they believed it is time wasting especially when they were trying to reach the deadline and there were lots of requests from customers and other testers, It caused a feeling for Brazilians that Poles act rude. Considering another cultural differences, when a member from Poland was working in Brazil to guide Brazilians, the way he gave feedback caused misconception. He collected them in a room and discussed about their work; what’s good, what’s bad and the possible improvement. Talking about their weakness in a group isn’t acceptable by Brazilians and that brought lots of challenges and controversy between them, as explained by the seventh interviewee. Eventually, a direct comment by E-mail from a tester setting in Poland to a developer in Brazil asking him to solve an issue, not in a complete
respectful and formal way, caused a challenging issue between them. Brazilians complained to line management in Volvo that testers in Poland don’t cooperate and after that Volvo IT management decided to do some investigation and have some team building practices to solve the issue. It all happened because cultural differences weren’t regarded well for that project. Also, as discussed in section 3.4.2.4, when there isn’t a personal bind between members and they don’t know the other members, they consider the distributed member as someone who should just do the tasks so misunderstanding and miscommunication occur for other parties’ mistakes. Also, when you don’t know someone, you don’t know the personality of other person so misconception might occur while communication. These issues show the necessary of considering them before it becomes an obstacle to team progress.

One reason that the problem occurred between developers in Brazil and testers in Poland was because they haven’t met before so they didn’t know about each other’s personalities and cultures. According to the seventh interviewee, before meeting Brazilians, they didn’t know that avoiding small talk and giving negative feedback when there are witnesses are considered offensive for Brazilians. When the mentioned problems happened, they decided to do some cultural investigation, put a travel budget for Brazilians to go to Poland, meet people there and have different workshops, presentations, gatherings, etc. which are parts of team building practices.

**Necessity of Team Building Practices**

As discussed in section 3.4.2.4, team building is the practice of creating a coherent team, an environment between different distributed teams that all members in different locations feel that they are working in one team with one goal. These practices are more useful to be considered from the start of the project not when problems started to show up as discussed in the previous section. The seventh interviewee, part of PBS team, believes that however it’s like a slogan that you need to know people you work with and meet them face-to-face, it doesn’t always consider seriously in Volvo. He continued that previously he had believed that team building activities is something non-necessary that was proposed by some people who had never did software development but after seeing the problems of not doing team building and seeing the benefits of regarding it, he admitted the advantages of it.

Focus on team building practices and the degree and level of it depends on project duration and team’s focus. Regarding quality aspect, it’s reasonable to invest only on degree of quality that is needed so there is no point focusing a lot on team building that are used a little. The cost of team building is another factor because however it’s always good to bring team and make them to get to know each other, if it’s not a long time perspective it might be harsh for the project budget. As a result, for smaller project that the objective is to do the job quickly; the dynamic of the team isn’t as much as concern for the project manager unless it becomes a problem. There is a belief in Volvo that it’s required to do team building if the project is critical and teams need to work together for a long time or there are some problems in hand or whether the teams works smoothly or not, as described by the seventh interviewee. As an example of a critical project, for one of the first distributed agile project in Volvo, which was an extremely tied deadline, fixed-price project with clear scope while the customer wanted to have a minimum scope of development and there was no flexibility in it, the whole team was co-located for a month. For this project, team members from Poland came to Sweden for three-four weeks to build the way they are
working because they needed to do everything right from the beginning, according to the first interviewee. It’s really rare that these kinds of comprehensive team building practice happen in Volvo.

As discussed above, however it’s suggested to have team building practices for longer projects, the project budget is now a factor for ignoring team building practices even when managers of teams are concerned about team’s being stable rather than looking out for new people. For the ALPHA project that main development team was located in India and the project took more than one and a half year, the initialization and team building practices was just coaching using some presentations and discussions between the offshore team members to know more about agile, planning, releasing and merging the code. There was just one architect from onsite location for guiding them through some domain level issues. Later and after five months, the architect and some other members came to India to give the team some training and, as the fifth interviewee mentioned, giving a proper picture of what they developed. In addition, the distributed meetings were by using Live Meeting tool, not using video, so there was no team building practices for that particular team. Even for the PBS project that they have a long perspective to assign the maintenance of the project to the Brazilian site and they are working for a long time with Polish site, there are many testers in Poland that has never met the Brazilian team before and they have never met the Swedish members before neither and as the seventh interviewee admits the cooperation between the testers and developers that have never met is more problematic than cooperation of testers who met these members.

One problem occurs by ignoring team building is the problem of communication. It’s mainly because members don’t know the proper way to address each other. Also, if you don’t know someone, you don’t care if he is good or not at work, if he is happy or not, etc. According to the seventh interviewee, many developers from other countries that aren’t traveling to Sweden are complaining that they don’t know each other very well so the communication is difficult for them. He believes that the communication within PBS team members is much better after team building practices and the collaboration between them works much better. Considering psychological issues, members respect each other more after knowing each other since they understand that they are real human beings not a name working in another country. In addition, team building practices have the advantage that members in different locations are more committed to each other.

On the other side, as interviewees 1, 6, and 7 mentioned and by considering the discussed problems of PBS team, teams will receive very limited output if project managers combine people and assign the roles in the team before they meet each other. For better understanding of the personality, and culture of your colleagues and to be closer to them and build interpersonal relationships, members of teams need to meet each other in one place to do some activities which aren’t related to work like going out, having drinks, joking, etc. It’s not related to the time-zone differences and even for teams working in similar time-zone, it’s necessary. That’s the best way of building a personal relationship but it’s not always possible. Below different ways of building a coherent team is discussed.
Traveling and face-to-face meeting

Using traveling for building teams depends on the budget, delivery time, location of distributed teams, etc. There was a routine for the first distributed agile projects to travel but then it changed. According to the first interviewee, for one of the first distributed agile projects in Volvo IT, which was between Poland and Sweden, the whole team was traveling from Poland to Sweden every third week mainly for Sprint Planning. One of the reasons for that team was delivery time factor since they wanted to build quality software from the beginning in a very uncertain environment with high-demanding customers so the teams needed to be on the same track from the very beginning. For projects, which are quite critical, and teams need to work together for a long time, this can be investment to be done at the start. As an important factor, if project has a long perspective, investing on traveling is necessary to reach consistency and avoid probabilistic future problems.

However the first interviewee believes that travel budget is more an investment rather than cost because teams are able to produce better quality software, it’s not routine for all projects in Volvo IT to travel. Traveling happens during different parts of the project. Meeting and collocating for first days of the project is critical for avoiding the problems from the start. Members collocating in one place and going out, having social gathering is required for building the personal bond relationship as confirmed by interviewees 1, 3, and 7. These are parts of the team building practices which members meet each other and discuss about things, which aren’t related to the project. As discussed in section 3.4.2.4, this gathering is also useful to discuss about the project and to make critical decisions about the project so members be more committed to the decisions. Also, it’s a proper time to have team exercise. These practices might be considered as costly but according to section 3.4.2.4 and as discussed with interviewees 1, 6, and 7, it’s good investment for enhancing quality and productivity.

According to section 3.4.2.4, using the first iterations for teams to get used to both technical and business knowledge of the project is a practice to avoid frustration in the middle of the project. It’s used to finalize requirements and develop a high level architecture. Preferably, whole teams should gather in one place but if it’s not possible because of cost, there should be some representatives of teams to have some level of face-to-face communication and to build trust and rapport. After that, they are able to transmit the captured knowledge to the members of their team. These visits are categorized as seeding visits.

Maintaining visits are also important for members not to stray from the right way of working, which is planned by project managers, and also to maintain the collaboration relationship. There should be a policy that members from different locations (both onshore and offshore teams) travel and stay in the other location for a while. The cycling traveling plan helps the team members not to get separated because of geography and time zone. The practice for a distributed project between Sweden and India was traveling two members from India to Sweden for two or four weeks to have interaction and discussion with Swedes and knowing them more, as described by the second interviewee. Then after 3-4 months, two members from Sweden were going to India for a while. It’s what recommended for the teams but it doesn’t happen always. For example, for the Indian team in ALPHA project, after not having any team building practices, there weren’t also too much maintaining visits and just one
architect and some developers went to India, explained by interviewee 3, 4 and 5. The developer gave them some training and showed what they have developed and what was going in the project. Even Scrum Masters in India don’t receive too much chance to travel to Sweden.

Some practices are recommended during the life cycle of maintaining visits to increase efficiency. Business knowledge transfer program is a practice used in PBS project after missing team building practices and after starting collaboration and communication problems between members of those teams. In this practice, there was always one experienced developer, either from Poland and Sweden to stay in Brazil for train them and answering the questions, according to the seventh interviewee. The problem with this practice was that using just one member wasn’t enough when he/she needs to deal with all developers, system analyst and architect. On the other hand, no time remained for him/her for producing and working on tasks. The ambassador practice as discussed in section 3.4.2.2, is another practice used for cultural, developing and business-oriented exchanges. The first interviewee believes that members in the project, who travel back and forth from different sides, and take part in Daily and Retrospective meetings and in general spend time with them, could act informally as ‘cultural ambassadors’. It’s possible since if you meet other sites, you bring some of your traditions so members of that sites could learn from it. Rotating team members is also necessary to achieve consistency between the sites. The practice recommends that at least one member be away from his/her main location for maintaining trust, getting along better and developing peer-to-peer relationship. Finally, sending experts to offsite locations are useful. These experts could be architects, project managers or members who have knowledge about business-side part of the project.

As discussed in section 3.4.2.4 and according to interviewees 1, 3, 6 and 7, some critical parts of the project are more important for members to travel. First, Sprint Planning meeting is an important period for teams to gather in one place but if it’s not possible for the whole team, Scrum Masters could join and work in one place. Also, User acceptance testing period is another important time. It’s when teams are trying to have a release and finalize it. In addition, it’s helpful that during major releases of the project, the team or at least some of members or representatives gather in one location. Finally, bringing the team together for the last couple of iterations makes releasing the final version of software smoother.

It’s suggested that different members from different sites travel. As discussed in section 3.4.2.4, Paasivaara, et al. (2008) suggests that customers and product managers should visit the development team and developers or developer representatives should travel to the customer site. It’s not a common practice in Volvo. For the PBS project, after adding the third team from Brazil, their members just came to Sweden in the first days of the project. After that, they decided that it’s not necessary that all members of Polish and Brazilian teams travel to Sweden so the only people that were travelling were the system analyst and the chief architect from Poland And one system analyst from Brazil. They travel on a regular basis, which is seven times in a year and usually two weeks each time. According to the seventh interviewee, these members have the best relations both with business side of the project and the requirement and development team in Sweden. This interviewee believes that the reason Brazilians don’t travel apart from to cost of travelling is jet
lag and time zone differences, which makes it hard for them to adjust to a new location. Developers in both sites are not traveling anymore. For the other projects, developers in India usually don’t travel to Sweden. The structure of teams in India is that they are mostly development teams and the most of the team are located there so it will be more cost effective that members of onsite location or other teams travel there. If teams have consultants, these experienced members also travel to guide the team. Sometimes business side people travel to the offsite location and spend some time with the team, usually couple of weeks to transfer some business side knowledge.

**Video and Cultural Tools**

Interviewees 4 and 5, working in India, believe that with having proper communication technologies and possibility of using videoconferencing, members aren’t required to travel especially members of development teams. It’s important to note that teams in India rarely use videoconferencing and have audio-only meetings. As discussed before, ignoring traveling and face-to-face team building practices brings cultural, communicative and quality problems but when it’s not possible because of the project budget, videoconferencing is an alternative. However, it doesn’t give the same feeling of meeting face-to-face, video is an alternative way for social gathering like celebrating a special holiday or a birthday, which is useful to build the team’s spirit even when members aren’t in one location.

Culture tools are also useful to have a guideline about other cultures. To have a general view about the culture of different teams and for guidance, a tool called Globe Smart is available in Volvo IT. It provides some particular matrix in terms of open communication, assertiveness, etc. and is categorized for certain points of each culture. For longer projects, they may invest in culture building using a team called culture and Backdoor. This team guides teams and works with the team culture in general, not a culture of specific country but more for culture of values, respecting the culture of other countries and respect for individuals so it’s more for generic culture building in the team. According to the third interviewee, not too many teams and projects use this team and facility. Usually, projects with big problems in the team culture or weak teams, which are working for a long time, invest on this particular practice.

### 5.2 Result Summary

1. What issues need to be evaluated before starting to use distributed agile software development method?

Before starting to use distributed agile method for a project, evaluations on both agile method and distributed development and comparing them with available resources of a project are required for checking the suitability of using the method. Regarding agile evaluation, it’s needed to analyze how much iteration and frequent deliveries are important for project managers and customers (Sutherland, et al., 2007; Tudor and Walter, 2006; Khan, et al., 2011). If customer provided comprehensive requirements and asked the team to develop it and he isn’t eager to work closely with the team, working on agile isn’t necessary for that project, as discussed by interviewees 1, 4, 6 and 7. Also, analysis on team composition and structure is important. According to
the forth interviewee, agile needs high skilled developers who are comfortable working in a team and are committed to the project, which is also confirmed by Cockburn (2002) and Pham and Pham (2011). Finally, if customers and users give feedback to the team and wrong development is avoided from the start, agile is considered as a cost effective way of development but hidden costs of it like cost of training members and user involvement should be regarded (see chapter 5.1.1.1).

Evaluating what companies gain and lose while distributing the project is the next step. According to interviewees 1, 3, 6 and 7, global companies need to reach other markets and required to have access to local market especially in the market that they are active and have manufacturing sites or services facilities. Also, they are able to reach global talents in other countries (Miller, 2008; Shrivastava and Date, 2010). On the other side, cost and productivity in a distributed development need more attention. By outsourcing to cheaper countries, it’s possible to reduce the labor cost but as discussed by interviewees 1, 6 and 7, the hidden cost of outsourcing like cost of traveling, team building, transitioning to a new location, etc. should be added to the total cost (Also confirmed by Sutherland, et al. (2007) and Miller (2008)). Interviewees 1 and 7 claim that distributing just because of the cost and not seeing other impacts of distribution is a mistake that is a reason for projects to fail. Productivity is also decreased when teams missed team building and have problems on communication and collaboration, according to interviewees 6 and 7 (see chapter 5.1.1.2).

The level of distribution is another challenge. Phalnikar, et al. (2009) believes that partial offshoring works better with shorter goals while complete offshoring is more appropriate for longer perspectives. Also, if teams are working in different time-zone, the level of dependency between teams should be decreased (referred to the project situations explained by interviewees 3, 4, 5, 6 and 7). Finally, if a distributed team is added to a project, that team should receive enough support and is required to have team building and knowledge transfer practices from the beginning of the project. By not following mentioned practice, that team will cause problems to the work of other teams in the project while they aren’t able to produce quality software, as confirmed by interviewees 6 and 7 (see chapter 5.1.1.3).

2. What are the main challenges while using distributed agile method and which practices are needed for coping with these challenges and problems?

After making decision to use distributed agile for a project, some challenges come up and some practices are needed to overcome it. When Scrum is used for a distributed project, it’s important that teams be cross-functional and have all needed skills (Miller, 2008; Pham and Pham, 2011). Also, the responsibility for a part should be distributed between locations so teams communicate more and solve problems together while when a member or a team isn’t available, teams could receive help from other locations, as discussed by interviewees 1 and 7. Pham and Pham (2011) claims that teams should be empowered to take decisions. Using catalysts, who are selected from team members and work closely with Scrum Master, is helpful to motivate teams to follow the planned method and is a proper way to empower members, according to the first interviewee. As another point, the role of Scrum Master is important in teams. According to interviewees 1, 6 and 7, the duty of Scrum
Master includes checking that teams are working in agile way of developing while making sure there is no obstacle for teams’ communication and collaboration (Also discussed in Miller (2008) and Pham and Pham (2011)). As a result, having a part-time Scrum Master causes teams not to work smoothly as he/she won’t be able to perform mentioned roles properly. Finally, if the number of Scrum teams working together and the level of interaction between them are high, there needs to be a distinctive role as Scrum-of-Scrum Master who handles the issues between teams (see chapter 5.1.2.1).

Scrum provides structured meeting for each Sprint. The problems with distributed meetings are availability of less time and necessity to have more time for arranging the meeting and discussing issues. For the distributed meetings, it’s important to choose a convenient time for all parties especially when time-zone differences are high (Deemer, et al., 2010). For Daily meetings, it’s recommended that teams discuss about what they plan to do today. Also, this meeting shouldn’t be just a status report to Scrum Master, as discussed by interviewees 1 and 7. For teams having distributed Daily meeting, pre-work before the meeting is required (Sutherland, et al., 2007). On the other side, it’s not always possible to have distributed meeting because of time-zone differences and because of the number of members locating in all sites, so teams need to have site-specific meetings (referred to interviewees 3, 4, 5, 6 and 7). These meetings have disadvantage that members just receive information about what people in their location are doing so they are required to check later if members in other locations aren’t interfering with their work, as discussed by seventh interviewees. After that, different Scrum-of-Scrum meetings between All Scrum Masters, project managers, business team and customer are required. Sprint Planning and Retrospective meetings are important for improving communication, coordination and team cohesion. In addition, Retrospective meeting is for discussing about obstacles and negative of positive points and for members who are not comfortable talking in a group, passing paper to write the ideas and reading them loudly is a good practice. Release Planning, Demo meeting and Sprint Review are other Scrum meetings (see chapter 5.1.2.2).

Distributed teams are working in different time zones and it brings some problems for team communication and collaboration. The problem mainly occurs when you need some one in another place but he’s not available and it becomes an obstacle for work progress. There are some practices to overcome it. First, Synchronization of working hours to increase the overlapped time is useful to have more communication between teams (Ramesh et al., 2006; Paasivaara, et al., 2008). Decreasing the dependency between sites is a useful practice to solve it (referred to interviews 3, 4, 5, 6 and 7). Another practice is to have someone in offshore location with same role for critical responsibilities like architect, as discussed the problem by interviewees 6 and 7 and suggested by Paasivaara, et al. (2008). For teams working in higher time-zone differences, it’s required that teams shift their work to have at least small overlapped time. Shrivastava and Date (2010) suggests that a solution to overcome it is using proxies or representatives in onshore and offshore locations, as the forth interviewee claims that a bridge gap or onsite coordinator role is used in Volvo. Finally, having one or some members to work early or late to answer the problems of other teams is recommended (referred to problem discussed by interviewee 6 and 7). This responsibility should rotate between members to avoid frustration (see chapter 5.1.2.3).
Poor communication has a negative impact on the quality, value and accuracy of the end result (Phalnikar, et al. 2009; Deemer, et al., 2010). In a DSD project, the level of communication decreases because of the distance but commitment from members and providing proper tools could mitigate this problem (Shrivastava and Date, 2010; Ramesh, et al., 2006). When teams are working distributed, they tend to work in silence thinking about their own part. As one of the duties of Scrum Master, he should encourage members to work together, connect people to solve problems together, create a communicate environment and makes sure that interaction works between members (referred to interviewees 1, 3, 6 and 7). Pair programming and code reviewing are two practices, which encourage members to communicate and work together while they have technical advantage to have a better quality software, as discussed in interviews 1, 6 and 7. For communication, it’s not recommended to use proxy because it leads to low bandwidth in communication while teams get dependent on few individuals. Also, information might get misinterpreted while transmitting the messages (see chapter 5.1.2.4).

Using appropriate tools are necessary for better communication. Discussed in interviews 1 and 3, video offers a better experience of online communication since some part of facial expression and body language is transmitted while makes it possible to see if someone has a question or is hesitant, happy, etc. Unlike audio-only meeting that people tend to be multitask during it, they have more concentration on discussion while using video (Deemer, et al., 2010). Video is also useful for team building practices and online and social gathering. According to interviewees 1, 6 and 7, E-mail is a tool for asynchronous communication but has the disadvantages of wasting more time and losing the mood of parties. Using proper issue tracking tools is also necessary. In addition, virtual and physical boards are required for good transparency and tracking project status. It’s required to simplify these tools to avoid wasting time updating them; regarding problems discussed by interviewees 6 and 7. Two other recommend tools are fist using a virtual board as a TV in each room and second equipping each room with videoconference tools to be used for both meetings and also passively (see chapter 5.1.2.4.1).

In a DSD project, people with different cultures are working in one project. These differences exist and should be considered from the beginning of the project before they cause problems to daily working routine, communication, collaboration and in general quality of the work (referred to interviews 6 and 7). Sutherland, et al., (2008) discusses that the solution to avoid these problems is concentrating on team building practices. It’s required because teams receive very limited output if managers combine people and assign the roles in the team before they meet each other, discussed by first interviewee. The degree of team building practices depends on project duration, project budget and the strategic goal of the managers. If the project duration isn’t short and if managers are looking for consistency within the team, they need to have comprehensive team building practices however the project budget affects the level of it (see chapter 5.1.2.5).

There are some practices for team building. Meeting and collocating for the first days of a project is critical for avoiding the problems from the beginning. As discussed by Ramesh, et al. (2006), Paasivaara, et al. (2008) and Miller (2008), this gathering is required to discuss about issues not related to the work for the aim of building
personal relationship. This period is also important for making critical decision by all members so they will be more committed to it. In addition, the first iterations is a proper time for teams to get used to both technical and business side of the project (referred to interviews 1, 6 and 7). After that, maintaining visits are necessary for members not to stray away from the right way of working and to maintain collaboration relationship. Members from both offshore and onshore location with different responsibilities should travel (Paasivaara, et al., 2008; Shrivastava and Date, 2010). For example, customers and product managers should visit the development teams and developers or developer representatives should travel to the customer site. Business knowledge transfer program, ambassador practice and rotating team members between sites are some recommended practices to make the best use out of maintaining visits (referred to interviews 1, 3, 4, 6 and 7). Using video and cultural tools are some less-effective ways of team building which is more appropriate for low-budget projects. It’s necessary to consider that these tools couldn’t be used as a full substitute of traveling (see chapter 5.1.2.5).
6 Discussion

This chapter provides the conclusion of this study based on the analysis provided in Analysis and result section. It also includes implications for the subject area and evaluations on method and results. This chapter concludes with ideas for continued research.

6.1 Conclusion

This study first covers evaluations for checking the suitability of using distributed agile development for software development projects and after that provides some suggestions for the challenges of following this method.

Having an evaluation about both agile method and distributed development and comparing the current situation and available resources of an organization with them is required to check the suitability of using this method. If customer needed frequent deliveries and is eager to work closely with the customer while he didn’t provide a detailed requirement, agile method is required for the project. Agile also needs high skilled members and however is known as a cost effective way of development, the hidden cost of using agile like cost of training and user involvement should be regarded. On the other side, however using DSD gives opportunity to reach global market and global talent and decreases the cost of development in the case of distributing to cheaper countries, it has some hidden costs like cost of team building and traveling and has the risk of decreasing productivity and quality of the work. Companies should also be careful with the level of distributing the work to offsite locations.

To have an effective agile distributed practice, some social challenges on using the method, which are mainly related to the selected agile method, time-zone differences, communication and team building, arise that needs proper considerations. Teams in Scrum, a trendy agile method, should be cross-functional and empowered to take decisions. Also, the role of Scrum Master to verify that communication and collaboration works within and between teams is important. Scrum suggested some structured meeting, which is useful for improving communication and collaboration between members. Sprint Planning and Retrospective meetings are recommended to be distributed but Daily meeting could be distributed or site-specific depending on the structure of the sites.

Time-zone differences cause problems for teams’ communication and collaboration. Synchronization of working hours, decreasing the dependency between teams, and increasing the balance between teams are some suggestions to overcome the problems. For teams with higher time-zone differences, assigning the responsibility to one or some members to work late or early and using proxies or representatives are required however using proxy in other situations isn’t suggested since it leads to low bandwidth in communication. Poor communication is another issue which has negative impacts on the quality, value and accuracy of the end result. Pair programming and code reviewing are two technical ways to force members to communicate more. Using appropriate tools are also necessary for better
communication. Using virtual and physical board, equipping videoconferencing facilities, e-mail, Wiki, etc., are recommended to increase the communication bandwidth and media for a distributed project. It’s important for teams to do their distributed meeting by using video since some facial expression and body language are transmitted. Finally, video provides a better environment for collaboration.

Cultural and team coherency problems could be solved by team building practices while ignoring it cause problems for daily working routine, communication, collaboration and in general quality of the work. The degree of needed team building practices depends on project duration, project budget and the strategic goal of the managers. Some practices are suggested for team building. Social gathering before starting the project is a way to build personal relationship and to make critical decision. In addition, the first iteration is important for teams to get used to both technical and business side of the project. After that, maintaining visits are necessary for members not to stray away from the right way of working and to maintain collaboration relationship. Members from both offshore and onshore locations having different responsibilities in the team should travel. Business knowledge transfer program, ambassador practice and rotating team members between sites are some recommended practices to efficiently use maintaining visits. Using video and cultural tools are some less-effective ways of team building which is more appropriate for low-budget projects. It’s necessary to consider the point that these tools couldn’t be used as a full substitute of traveling.

6.2 Implication for the subject area

Three different groups could use the results and findings of this report:

1. Volvo IT Company Employees as the main target of empirical study.

My interviews with employees who work in different sections of Volvo IT Company helped me to gain an overall view about the situation, challenges, and practices of distributed agile projects in the company. Also, the data provided in Empirical Study section of this research and the comprehensive analysis, which was performed in Analysis and Result section of the report, could help them to understand the positive and negative points of their practices to help them improve it.

Different groups within Volvo Company could use this report based on the information they need. First, project managers who are eager to use Agile, DSD, or Distributed Agile for their project could consider the results of this dissertation project. The important part for them is the results of first research question where I aim to give an evaluation about problems and also opportunities to use these methods. Second, I target project managers and Scrum Masters who are currently active in distributed agile projects. It’s more important for them to consider the second research question and to review the data provided in Empirical and Analysis and Results sections. By using it, the shortages of current practices are better understood and suggestions provided by analysis help them to have a better practice. Finally, members of distributed agile teams in Volvo IT could use it to capture an overview
about challenges and needed Agile and Scrum practices. They could use both research questions and the data provided in theoretical, empirical and analysis Sections.

2. Organization using distributed agile development method

The results of this study could be used for companies interested in distributed agile method especially Western companies who are using this method or planned to distribute their development to other countries especially to countries that are located in different time-zones and have different cultures. They could use the data provided for the first research question in theoretical, empirical and analysis sections to evaluate and compare the resources of their company with data provided in these sections. The information for second research question provided in this report is helpful to understand the ways and practices to regard the mentioned challenges.

3. Researchers in the field of Informatics

Researchers within the field of Informatics specifically researchers who want to gain more knowledge about agile methodologies and distributed software development could use this dissertation project. It provides a basement for them to understand the forces, opportunities, challenges and needed practices of distributed agile development. Finally, it gives knowledge to students who are interested to agile methodologies, distributed development or combination of these methods as trendy development ways which will be used more and more for software development.

This research followed theoretical procedures to make it usable as a reference for researchers and practitioners who are active in this research area.

6.3 Method evaluation

In the theoretical section of my study, first, I read and analyzed different texts to understand the current situation in my research area. Since I aimed to cover the challenges and practices of Volvo IT, I also needed to cover literature resources, which were explaining the previous implemented cases in other global companies using distributed agile method. As I had previous researches related to agile methodology, I had an overview about it but I needed to search about Scrum method and I needed more information about DSD development. Analyzing these theoretical resources helped me to divide the second research question into four different sections and cover the problems and practices related to Scrum, time-zone differences, Communication and collaboration and finally team building. Since the area of distributed agile development is wide, I aim to mainly cover the social aspects of Agile distributed development from the start of the project.

For the empirical chapter, I performed interviews with employees who are active in distributed agile project. The distribution of my interviewees helped me to cover members with different roles that are living in various regions and have different cultures. As a result I had interviews with Scrum Masters, developers, lead developers, leaders and architects who are working in Sweden, Poland and India. I also used different communication method for interviews consisting face-to-face discussion, online video interview and audio-only interview. Before interviews, I sent
the questions to interviewees that helped them to understand my research and be prepared for the questions.

I used semi-structured interview, which gives me the opportunity to deeply explore the discussed issues. During interviews and when an interviewee was discussing about technical challenges, I needed to change to direction of the interview to social factors by more explaining the question since I aimed to cover social challenges of following distributed agile method. Some interviews gave me more knowledge about the method that forced me to review some more literature resources to understand better the theoretical factors of the issue.

6.4 Result evaluation

Validity is related to the integrity of findings and the conclusions come from a research. Some factors are regarded for evaluating the result. Consistency between the research question and the data captured in theoretical and empirical chapters and also the methods for gathering the data is the first important factor. First of all, I cleared the goal of my research and clarified the method for analyzing the theoretical data. By analyzing the theoretical data, I divided my research questions into two sub questions and even for second sub question, I made it clear that I want to focus on four factors: challenges and practices of Scrum, time-zone differences, Communication and collaboration and finally team building. After that, I used interviews to verify the result of my theoretical study. Finally, I divided the analysis part of the research into two sections: first section is comprehensive analysis of theoretical and empirical data and the second section consists answering research questions by summarizing the analysis of the first section.

There is another criteria for evaluating the result of a study that are explained by Guba and Lincoln (1994). The first factor of it which concerns about validity of a research is credibility, as discussed in chapter 2.6. This factor concerns about receiving confirmation of the content and result of the research from the members of the social world the research is investigating. To reach this criterion, the data captured in the theoretical study of this research was verified by different members who are currently active in distributed agile development project. These members consist wide range of members who have different roles and are working in different places. Transferability is the next factor, which entails generalizing the results across social settings. Because of the nature of qualitative study that focuses on a specific group, the generalization of this method is a challenging issue. In my research, I analyzed agile distributed method for a western country that aims to distribute their development to cheaper countries so the challenges and problems could be adapted to a similar situation. Dependability, the next factor, which is another term for reliability, discusses that whether the results of a study are repeatable. Because of the nature of my work as a qualitative study and because I interviewed with specified people this criterion is reachable in my research. Finally, I analyzed the data captured in theoretical and empirical part of my study in a way that no personal values impact the phases of the research.
6.5 Ideas for Continued Research

There are some areas in my research that I wasn’t able to focus on it so it gives an opportunity to other researchers to focus on it. First of all, I couldn’t have observation method in my research especially observing how works going on in a distributed location. The suggested area of focus could be how members communicate with other teams’ members and what problems happen when a member of other location isn’t available.

Another idea to work on it is considering the technical factors of agile distributed method, the quality of the work and also analyzing the integration and documentation problems. Analyzing the tools for a project is also useful for researchers to focus. There are many tools and each one has some strength and weaknesses and finding the proper tools for projects by considering the budget is worth for a research.

Finally, Seeing the impact of collaboration with end-users for a distributed agile project is an idea for further research. A quantitative research is also helpful to check how end-users are satisfied with current collaboration and product.
7 References:


Cockburn, A. Agile software development. 2002.


Morgan, R. E. (2008). *Distributed AgilePlanner: A card-based planning environment for agile teams*. (Dissertation/Thesis), ProQuest, UMI Dissertations Publishing. Retrieved from [http://hb.summon.serialssolutions.com/link/0/eLVHCXMwVV0xDsldwDLsrgA2KHrDvAoiZN3XpEQNUBtg6siWOLGcH_SQV1MPlu8nKvzssdwCaxOm1RKGl72etYezZMsU2ayfhvbi55v0Kxv40Hoby0wNQ3pB8SayhkSwVyzKFn8c4peQxiXTMVcoOA DsO6NhWyt6Kt45a0AaMqWYyYu4Yiyv9JioHjcn3nMXGXYrGOLKw1Hvz9f3nT5pbtgYF5v_QtfhC83](http://hb.summon.serialssolutions.com/link/0/eLVHCXMwVV0xDsldwDLsrgA2KHrDvAoiZN3XpEQNUBtg6siWOLGcH_SQV1MPlu8nKvzssdwCaxOm1RKGl72etYezZMsU2ayfhvbi55v0Kxv40Hoby0wNQ3pB8SayhkSwVyzKFn8c4peQxiXTMVcoOA DsO6NhWyt6Kt45a0AaMqWYyYu4Yiyv9JioHjcn3nMXGXYrGOLKw1Hvz9f3nT5pbtgYF5v_QtfhC83).


Shrivastava, S. V., & Date, H. Distributed Agile Software Development: A Review.


University of Borås is a modern university in the city center. We give courses in business administration and informatics, library and information science, fashion and textiles, behavioral sciences and teacher education, engineering and health sciences.

In the School of Business and IT (HIT), we have focused on the students' future needs. Therefore we have created programs in which employability is a key word. Subject integration and contextualization are other important concepts. The department has a closeness, both between students and teachers as well as between industry and education.

Our courses in business administration give students the opportunity to learn more about different businesses and governments and how governance and organization of these activities take place. They may also learn about society development and organizations' adaptation to the outside world. They have the opportunity to improve their ability to analyze, develop and control activities, whether they want to engage in auditing, management or marketing.

Among our IT courses, there's always something for those who want to design the future of IT-based communications, analyze the needs and demands on organizations' information to design their content structures, integrating IT and business development, developing their ability to analyze and design business processes or focus on programming and development of good use of IT in enterprises and organizations.

The research in the school is well recognized and oriented towards professionalism as well as design and development. The overall research profile is Business-IT-Services which combine knowledge and skills in informatics as well as in business administration. The research is profession-oriented, which is reflected in the research, in many cases conducted on action research-based grounds, with businesses and government organizations at local, national and international arenas. The research design and professional orientation is manifested also in InnovationLab, which is the department's and university's unit for research-supporting system development.