Master Thesis
Software Engineering
Thesis no: MSE-2011:51
June 2011

Hindrances for Agility: Detection and Recommendations

A Case Study on Software Process Improvement in a Globally Distributed Environment

David Musat

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of European Master in Software Engineering. The thesis is equivalent to 24 weeks of full time studies.

Contact Information:
Author: David Musat
Address: Kolonivägen 1C, 37154, Karslkrona, Sweden
E-mail: davidmusat@gmail.com

University advisors:
Dr. Darja Šmite
School of Computing
Blekinge Institute of Technology

Dr. Óscar Dieste
Facultad de Informática
Universidad Politécnica de Madrid

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

Internet: www.bth.se/com
Phone: +46 455 38 50 00
Fax: +46 455 38 50 57

ABSTRACT

Context. Global Software Development is software work undertaken at geographically separated locations across national boundaries in a coordinated fashion involving real time or asynchronous interaction. Distributed Agile Development aims at the benefits of both Agile Software Development and Global Software Development aiding the distributed teams to overcome the challenges brought by the distribution.
Objectives. In this study the author investigates whether a globally distributed company is prepared to be agile, determining hindrances for agile and providing recommendations to mitigate or overcome the detected hindrances.
Methods. In this case study, surveys and interviews were used to study the hindrances for agile and literature was used to provide the recommendations towards the detected hindrances.
Results. 4 hindrances were detected. Only 1 was justified as necessary for the good performance of the distributed company. Several recommendations to overcome the hindrances were proposed. Both hindrances and proposed solutions were validated by the company representative.
Conclusions. We conclude that the studied individuals are willing to be agile. As agile is built bottom-up, the company is prepared to be agile. However, they will not be able to be agile until they overcome or mitigate the detected challenges. In the study, several solutions for it are proposed.

Keywords: Agile Distributed Development, Case Study, Hindrances, Recommendations

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TERMINOLOGY

Term/Abbreviation	Definition
Author	Student responsible of writing this thesis (David Musat)
Challenges	Challenges and hindrances are used interchangeably in this thesis. Issues that negatively affect the agility of the development process model.
GSD	Globally distributed Development (Also known as Globally distributed Engineering)
ASD	Agile Distributed Development. ASD, Agile Practices and Agile are used interchangeably in this thesis.
DAD	Distributed Agile Development – An approach that takes advantage from both ASD and GSD
Company	The company in which this case study was performed
RCA	Root Cause Analysis – Process designed for use in investigating and categorizing the root causes of events
PdM	Product Manager
UM	Unit Manager
PMPAT	Company's Project Management Process for Agile Team specification
Dev	Developers

1 INTRODUCTION

1.1 Background

1.1.1 Global Software Development

1.1.1.1 Definition

Sahay defines Global Software Development (GSD), also known as Global Software Engineering (GSE), as software work undertaken at geographically separated locations across national boundaries in a coordinated fashion involving real time or asynchronous interaction [1]. Globalization of business has dramatically impacted the software industry. Today, more and more projects are distributed among several geographically dispersed locations. This is making GSD become a norm in the software industry [2].

Assumed benefits of GSD include reduction in salary related costs, cycle time arising from follow-the-sun and access to a larger range of skilled experts [3]. In GSD the challenges are present in terms of communication, coordination and control [4] caused by the geographical, temporal and socio-cultural distance.

Developing software in the distributed ways implies that teams involved are also distributed among different locations. The distribution can be done in several ways [1]:

- **National Insourcing**: team members from the same company working in different locations in the same country.
- **National Outsourcing**: team members from different companies working in different locations in the same country.
- **Offshore Insourcing**: team members from the same company working in different locations from the globe.
- **Offshore Outsourcing**: team members from different companies working in different locations from the globe.

The study in this thesis is focused on exploring offshore and national insourcing arrangements in a software company in Sweden working distributedly with its subsidiaries in Sweden and Russia, and offshore outsourcing covering the company's links with a sub-contractor in Denmark.

1.1.1.2 History

Global Software Development has been evolving since past 15 years. GSD started when India began to transfer entire capabilities to the West in a whole piece [1]. However, in an industrial context, IBM was a pioneer in this form of work [5].

Since 1980 outsourcing has been incrementing across national and cultural borders [1]. Cost has always been the major driver to adopt this approach, especially when there are low wage countries involved in the development process [1].

GSD became popular in the early 1990s and researchers and practitioners began to document their experiences and approaches [1]. Currently, articles reporting best practices and approaches coming from GSD case studies have become a trend. It helps practitioners to understand the challenges they face. However, the field is still immature and more empirical studies reporting best practices are required [6].

1.1.1.3 Challenges of GSD

Practitioners have realized that the application of GSD is more challenging than even the most complex project managed entirely in house [7]. As Smite et al. [6] state "there is still no recipe for successful and efficient performance in globally distributed software engineering". These challenges can be grouped in three main categories [3]:

- **Socio-cultural Distance:** is a measure of an actor's understanding of another actor's values and normative practices [8]. Human-related issues have a big impact in team working [9]. In the case of GSD, culture has a big effect on how people interpret other people's actions. In this dimension religion, national language, politics and ways of behavior among others can affect productivity [9,10].
- **Geographical Distance:** Geographical distance is a measure of the effort required for one actor to visit another and can be seen as reducing the intensity of communication [8], especially if one of the locations have problems with media and cannot substitute face-to-face meetings.
- **Temporal Distance:** Temporal distance is a measure of the dislocation in time experienced by two actors wishing to interact [8]. Time difference can affect on the distributed team by reducing opportunities to establish collaboration [1,10].

Socio-cultural, geographical and temporal distances bring challenges to GSD that project managers need to be aware of [11,12]. The success of a distributed project highly depends on the mitigation of these challenges. The selection of a software life cycle model also has a significant impact on the success or failure of the project itself [13], thus it is important to understand the role of life cycles in GSD as well. Agile software development is a software life cycle model that seems promising in the mitigation of the aforementioned GSD challenges.

1.1.2 Agile Software Development

A software life cycle model or process model refers to how a project is planned, monitored and controlled from the beginning until the end of the software product [13]. There are different software life cycles, going from the traditional (Waterfall [14]) to the modern ones (Agile Software Development [15]).

Agile Software Development (ASD) is a term coined when the Agile Manifesto [15] was formulated in 2001. It refers to an iterative software life cycle model where requirements and solutions are satisfied through self-organizing cross-functional teams. ASD is an approach that is becoming very widespread and has a huge impact on distributed software development. This is caused by the improvements that agile practices introduces compared to the traditional model practices [16]. While traditional methods rely on predictability, agile methods rely on adaptability. Agile methodologies also try to reduce overhead in justification, rationale, meetings and documentation, keeping them as low as possible.

Distributed Agile Development (DAD) takes advantage of both ASD and GSD. DAD aims at the benefits of both ASD and GSD [17] aiding the distributed teams to overcome the challenges brought by the distribution. Contrary to ASD, GSD typically relies on formal mechanisms. Despite this, many companies bet for the combination of ASD and DSD. As a result, they attempt to both approaches into a common DAD software life cycle model [18].

Several case studies have fetched new data for studying how GSD could benefit from DAD:

- Authors from [19] report a case study built within three different companies. The objective was to understand the difficulties faced in managing a DAD development process and the practices designed to address them.

- The study covered in [20] reflects an identification and classification of the main problems faced in DAD. It was performed selecting DAD literature related case studies. Almost all of the data included in the study come from case studies related to Scrum and XP in onshore and offshore teams.
- An Irish Intel case study revealed which of the assumed benefits of GSD are currently being achieved with the application of DAD [3].
- Two different case studies performed by Yahoo! [21] in the development of a news reader gadget and a podcast system showed how some Agile teams success and others fail. It also provides guidelines and best practices to avoid failures in the application of DAD.
- Some advises and best practices about adopting DAD can be found in the conclusions extracted from the case study presented in [22].

DAD needs are wider since it has to cover the GSD and ASD necessities. There are examples of success and failure. As it can be seen, there is a growing trend towards balancing agile in distributed approaches to meet the challenges of communication, control and trust [3,23] since DAD seems promising to overcome the GSD challenges.

1.2 Motivation

Only 11 out of 59 studies actually evaluating a particular method, technique or tool for GSD were detected in a systematic review published in 2010 [6] and just a 2.5 of them were framed in an industrial context. As Smite et al. [6] state "there is a lack of studies conducted in industrial environments that at the same time investigate a particular method, practice or aspect of software engineering knowledge areas". Specifically, there is a lack of studies related to tools and process models (merely 3% of the topics of investigation addressed in literature were related to this field) [6].

In addition, only 20% of the studies retrieved during a systematic review published in 2009 [24] to collect the available tools, best practices and available process models reported in literature are from industrial context. As Da Silva et al. state [24] "It is clear from these findings that although the number of studies is increasing, there is still the need for more empirical research to create stronger and quantifiable evidences of the effect of best practices, tools, and models on DSD". As a result, more research to evaluate different practices, methods and techniques rather than mainly focus on managerial problem-oriented lessons learned is required [6].

To address the challenges caused by the software business globalization, several globally distributed companies have started looking at how to become agile while being distributed, [22,25,26] are examples of it. The company involved in this thesis work is willing to be agile. The scope of this thesis includes studying the current and desired collaboration between distributed locations of the company and agility of the software development processes, detecting hindrances for agility and providing recommendations on how to overcome these hindrances.

The findings of this thesis are relevant for:

- Globally distributed software companies that want to be more agile. The research methodology of this thesis could serve for them to determine if they are prepared to be agile.
- Globally distributed software companies that are facing challenges while applying agile. The detected challenges and proposed recommendations can serve as guidelines to overcome their hindrances for agility.

1.3 Aims and Objectives

The main aim of this thesis is to study whether a globally distributed company is prepared to be agile, determining hindrances for agile and providing recommendations to mitigate or overcome the detected hindrances. The aims and objectives of this thesis are addressed in the form of a case study carried out in a globally distributed software company specialized in developing embedded systems and software applications.

- **Aim 1**: To determine if the company is prepared to adopt agile development.
  - **Objective 1.1**: To determine whether the current state of the development process satisfies the needs of the agile principles.
  - **Objective 1.2**: To determine the hindrances for agility in the development process.
  - **Objective 1.3**: To seek for the root causes of the potential hindrances for agility in the development process.
- **Aim 2**: To provide recommendations based on the best practices reported in the GSD literature for the overcoming of the root causes.
  - **Objective 2.1**: To determine the available best practices suitable for helping the company to overcome the detected root causes
  - **Objective 2.2:** To determine the available tools that can enhance the agility of the development process.
  - **Objective 2.3:** To validate the suitability of the proposed practices with the company's representatives.

1.4 Research Questions

R.Q. 1 Is the globally distributed company prepared to be agile?

- R.Q. 1.1 Does the agility of the development process satisfy the needs for agile principles?
- R.Q. 1.2 What are the hindrances for agility in the development process?

R.Q. 2 What the root causes of the detected hindrances for agility are?

R.Q. 3 What recommendations can be given to address the detected hindrances' root causes?

- R.Q. 3.1 What are the available best practices that could help to overcome the root causes?
- R.Q. 3.2 What are the available tools that could help to overcome the detected root causes?
- R.Q. 3.3 What is the suitability of the proposed recommendations?

1.5 Research Methodology

1.5.1 Empirical Context

1.5.1.1 Project Background

The work reported was carried out within R2D2. R2D2 is funded by Ericsson Software Research and the Swedish Knowledge Foundation under the KK-Hög grant 2009/0249 [27]. Its focus is the construction of a framework for supporting offshoring decisions including

techniques for evaluating potential benefits, risks and long-term effects for different offshoring scenarios, and guidelines for planning and building successful collaborations under the following slogan: *"Decision Support for Offshoring Software Development"* [27].

1.5.1.2 Company Overview

The case study was conducted within a globally distributed company. It is a diversified global manufacturing and technological company. The company is focused on a wide range of products and services in the industrial, commercial, and consumer markets through its network power, process management, industrial automation, climate technologies, and tools and storage businesses[1].

The company's headquarter are located in U.S.A. but it makes business in more than 150 countries. It has offices in 240 different locations, from which approximately 160 are located outside U.S.A. It also has had a long-time R&D, manufacturing and sales presence in Western Europe, and a growing presence in Eastern Europe and Russia.

Eastern Europe offers particularly high growth potential for this company. It has been an active investor in this region, where the company has established advanced engineering design facilities and a number of manufacturing sites. The case study covered an investigation of the work undergoing in the offices of the company located in Sweden, Denmark and Russia.

The studied products are developed by teams that approximately sum up twenty developers and testers located in Russia, and thirteen product owners and very few developers located in Sweden. No reliable information could be extracted about Denmark and the distribution between other locations.

1.5.2 Case study methodology

Several well known software process improvement (SPI) frameworks exist, most of them cyclic with four main steps: an evaluation of the current practices, planning for improvements, implementation of the improvements and an evaluation of the effects of the improvements [28]. There are two types of SPI frameworks: inductive and prescriptive [29]. Inductive SPI frameworks such as the quality improvement paradigm (QIP) [30], are based on understanding the current situation and basing improvement efforts on addressing the most critical issues. However, prescriptive models such as CMMI [31] and ISO/IEC [32] do not take into account the organization's specific needs.

We can say that the SPI needed framework for this thesis was inductive since the author knew the proposed improvements are based on the study of the current organization's state. Finding possible improvement issues based on organizational needs, and not by following a prescribed framework, can help assure support from practitioners and management alike for the assessment and subsequent improvements [28]. For this reason, the author did not follow any prescribed framework but designed the study in a flexible way adapting it to the encountered challenges.

The company studied is based in Gothenburg, Sweden. The thesis followed a case study research methodology according to the guidelines from [33]. Taking into account these descriptions, explanations and definitions it can be concluded that:

- The purpose of this case study is a combination of both explanatory and improving. It is explanatory since an explanation of a situation or a problem is sought. As a result the

[1] This description is taken from the company's webpage. The citation is not provided to preserve the company's anonymity.

hindrances for agility in the development process the company follows were investigated. It is also improving since advices and suggestions to mitigate the detected challenges for agility were provided.

- The research perspective is interpretive because the study is based on extracting information from the understanding of the participants' interpretation of their organizational context, namely the organizational context.
- Data extraction was both qualitative and quantitative. As authors from [33] states *"a combination of qualitative and quantitative data often provides better understanding of the studied phenomenon".*
- The selected research process was flexible. It had a protocol, but new approaches and tools were introduced to overcome the challenges.
- The extraction of information was triangulated in both research perspectives and research methods:
    - In the first case the information was triangulated in different roles, i.e. developer, unit manager and product manager (see Figure 1a). The roles were selected from the organizational mapping the company provided. Developers were selected to determine if the agile adoption was possible since it is built bottom-up. Product managers were selected to determine the satisfaction of the internal client with the current practices. Finally, the unit manager was the representative of the company that had an overview of all the studied products allowing making comparisons between the retrieved data and his perspective.
    - In the second case the information was triangulated in different research methods, i.e. surveys, interviews and literature (see Figure 1b). Surveys were selected to determine the current state of agility of the development process. Interviews were selected to make a root cause analysis and to contrast the retrieved data from the surveys with the opinion of individuals. Finally, literature review was selected to provide recommendations based on well proven best practices and available tools.

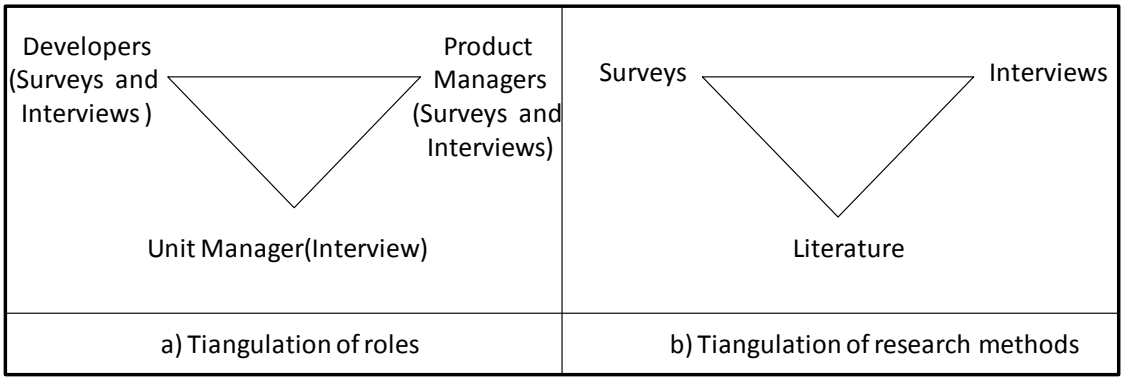

Figure 1. Triangulations in the case study

1.5.3 Research Design and Execution

The company of this case study is attempting to adopt agile in the large. For this reason, the representatives of the company and the supervisor of the thesis put forward a task for the author of the thesis to determine the agility of their development process, to research which the hindrances for becoming agile were and to provide suggestions and recommendations to mitigate the detected hindrances.

1.5.3.1 Surveys

A survey is a comprehensive system for collecting information to describe, compare or explain knowledge, attitudes and behavior [34]. Survey data represents the current situation of a studied event. In this thesis, two exploratory surveys were designed and conducted for detecting the current and desired level of agility of the company's development processes.

The survey method was selected to determine the perspectives in terms of the collectives formed by developers and product managers. The two surveys were role-focused to contrast both perspectives. The surveys' design and analysis were based on the guidelines from [35]. The questionnaires and data were designed, monitored and gathered using Google Docs Form Creation web-based tool [36]. Data was analyzed using IBM ® SPSS Statistics 17.0 ® [37] and Microsoft Excel 2007 [38].

The surveys were composed with the knowledge acquired on a literature review that was performed in advance to understand the usual process models followed in GSD [39], detect the most common role distribution in DAD [40] and study the agile principles [41]. The company also provided their corporate guidelines for implementation of agile practices called Project Management Process for Agile Teams (hereby PMPAT). The population of the surveys was subjects involved in globally distributed projects.

The main purposes of the surveys were:

- Capture the current agility of the software development processes followed by the distributed company.
- Capture the desired agility of the software development processes followed by the distributed company.
- Capture satisfaction and dissatisfaction points of practitioners with the current software development processes.
- Capture the tools used in a daily routine.

The survey contained lists of tools and roles based on related literature [40,42] as well as questions regarding the use of agile practices based on [41]. The corresponding questionnaires were forwarded to product managers and developers. The results of the surveys were used to identify problems in the development process. There were some open questions forming the questionnaire so additional problems and suggestions for improvement were collected and included in the study.

The survey revealed some hindrances for the company to become agile. To provide more suitable solutions to address the detected challenges, a Root Cause Analysis (RCA) in form of interviews was performed.

1.5.3.2 Root Cause Analysis

Root cause analysis (RCA) is a process designed for use in investigating and categorizing the root causes of events with safety, health, environmental, quality, reliability and production impacts. In other words, RCA helps to identify what, how and why something happens, thus preventing recurrence [43]. In this thesis, the RCA is interview based.

In interview-based data collection, the researcher asks a series of questions to a set of subjects about the areas of interest in the case study. Interviews serve to go deep in the research asking an individual interview guided questions [33]. Interviews were used to identify the root causes of the hindrances detected in the surveys. These interviews for the RCA were designed following the guidelines from [43-45]

The interview method was selected to know the feeling of two individuals related to the findings of the study. The interviews were semi-structured combining open-ended and exploratory questions. They were conducted face-to-face and designed in an incremental way, i.e. the interviews questions were formulated referring to the data gathered in previous steps of the research methodology.

Two interviews were conducted. To take into account the distribution and the different perspectives the interviewees came from two different locations and were performing two different roles in software distributed projects, i.e. a unit manager from Sweden and a developer from Russia. The interviews revealed some of the root causes of the studied challenges. However, only two interviews were performed and there is a chance that some other root causes were not identified. Therefore, the interviews were beneficial and essential on detecting where, why and how the hindrances were originated. The interviews also helped the author with some ideas for addressing the mitigation of these hindrances.

1.5.3.3 Literature Review

Once the root causes were detected, the recommendations to mitigate the detected hindrances could be provided. A systematic literature review was considered to be needed for this last stage. The objective was to collect all the tools, process models and best practices reported in literature suitable for addressing the mitigation of the detected hindrances. A systematic literature review has three steps: planning the review, conducting the review and reporting the review [46]. The author of this thesis decided to take the first step of a systematic review protocol following Barbara Kitchenham's guidelines [46].

When performing the planning step (identification of the need for a review and development of the review protocol), some related systematic literature reviews were discovered. After checking their validity, consistency and suitability, one of them was selected for the last stage of the research methodology. As a result, there was no need to perform a separate systematic literature review.

The selected systematic review had some completeness constraints that were studied and documented. Despite these, the systematic review was still considered suitable for the assessment and was one of the milestones of this study.

1.5.3.4 Company Representative Survey

With the recommendations and hindrances already studied, the author of this thesis wanted to evaluate whether the detected hindrances were unknown by the company and if the proposed solutions were useful and intended to be implemented.

A survey was designed with the purposed of evaluating the novelty and the usefulness of the thesis. The respondent (company representative) was shown the detected hindrances and the proposed solutions and asked for his opinion towards the results of the study.

1.5.3.5 Mapping of the Research Questions to the Research Methodology

Each research question can be mapped to the steps and methods explained in the research methodology. Table 1 depicts the mapping of the research questions into the different stages of the research methodology in which they were studied. Figure 2 shows the graphical representation of the research steps.

Research Question	Sub-Research Question	Stage of the Research Methodology
R.Q. 1	R.Q. 1.1 Does the agility of the development process satisfy the needs for agile principles? R.Q. 1.2 What are the hindrances for agility in the development process?	PdM and Developers Surveys
R.Q. 2	—	Root Cause Analysis and Literature Review
R.Q. 3 R.Q. 3	R.Q. 3.1 What are the available best practices that could help to overcome the root causes? R.Q. 3.2 What are the available tools that could help to overcome the detected root causes?	Literature Review
	R.Q. 3.3 What is the suitability of the proposed recommendations?	Company Representative Survey

Table 1. Mapping of the RQ to the Research Methodology Steps

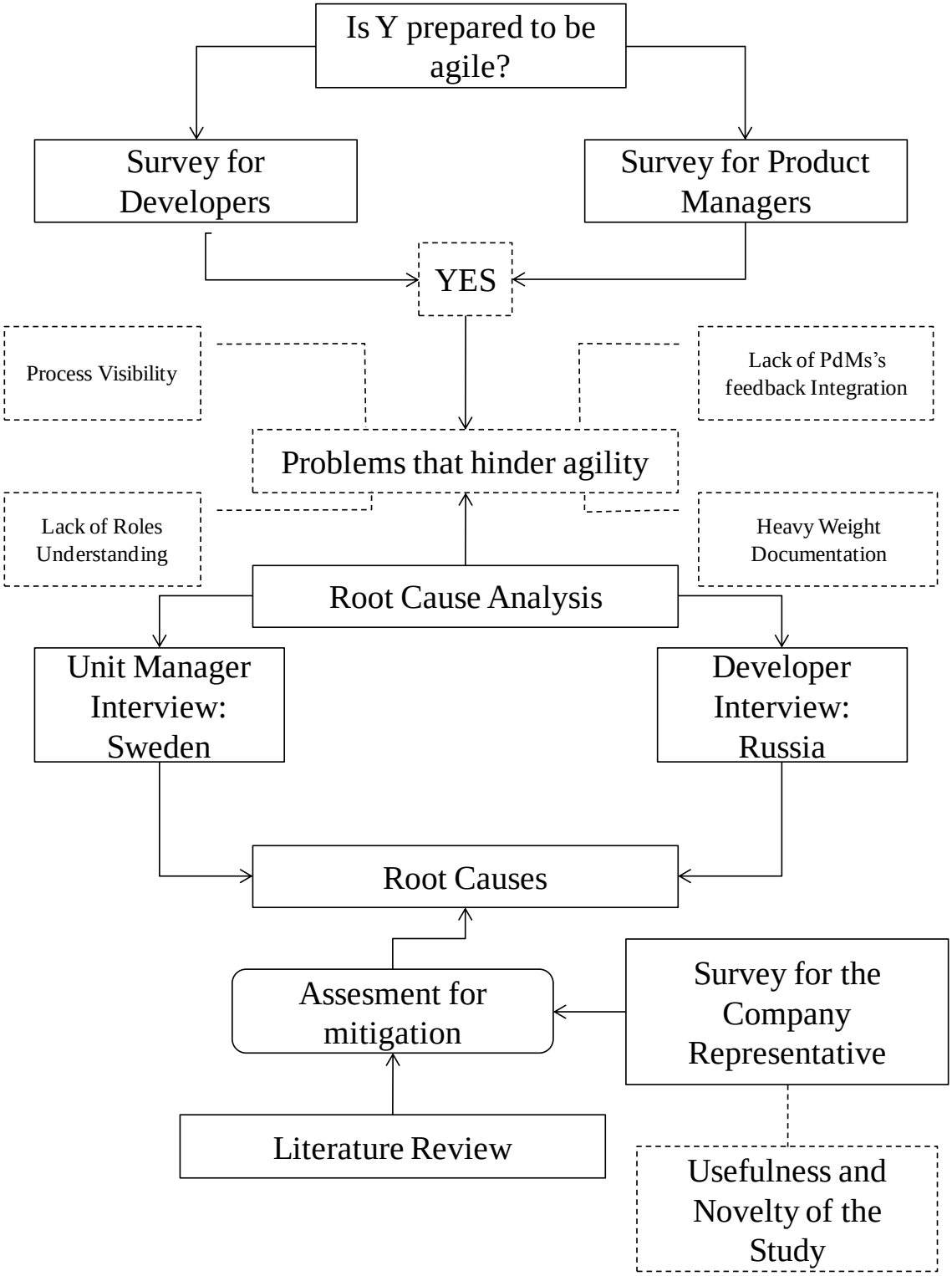

Figure 2. Research Steps

1.6 Research Outcomes

This thesis is a part of a research project that aims to provide a framework for supporting offshoring decisions including techniques for evaluating potential benefits, risks and long-term effects for different offshoring scenarios, and guidelines for planning and building successful collaborations [27]. Contribution of this master thesis which are as follow:

1. Provide empirical data that determines whether or not the company is prepared to benefit from the agile practices.
2. Determine existing risks for agility in the software development process based on the extracted empirical data.
3. Study the root causes of the detected risks.
4. Provide guidelines based on best practices and tools to overcome the detected root causes.

Outcomes 1, 2 and 3 are to be achieved by conducting a case study in a globally distributed software company. Outcome 4 is to be achieved with literature.

1.7 Structure of the thesis

Figure 3 shows the structure of the thesis, which has three main categories.

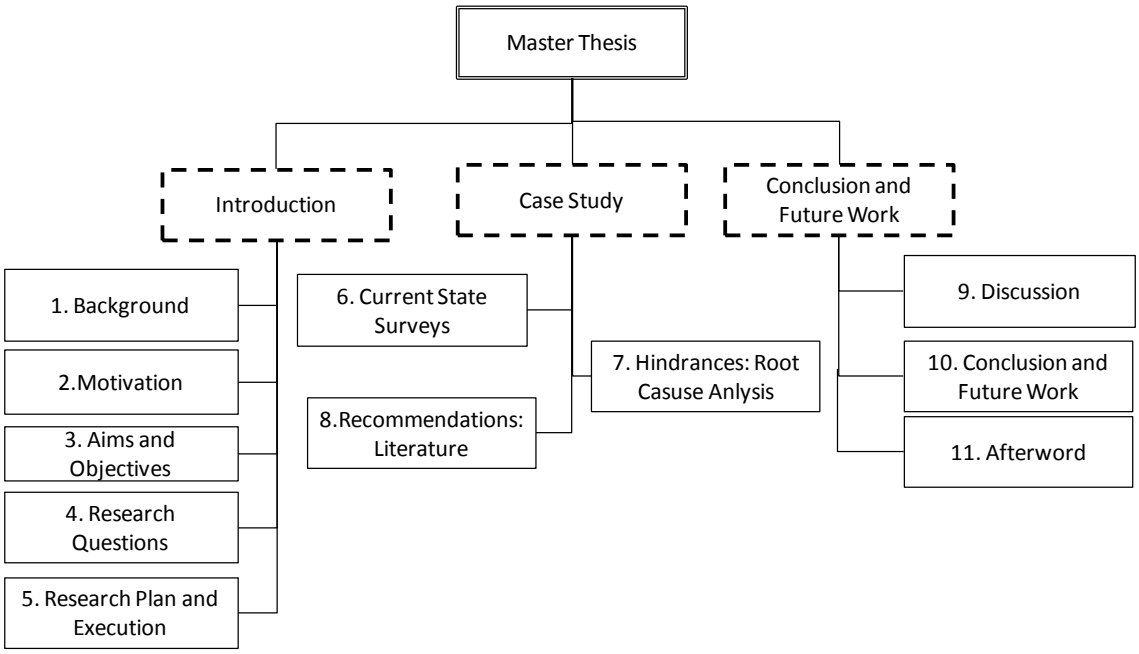

Figure 3. Thesis Structure

2 PREPARATION FOR AGILITY: CURRENT STATE SURVEYS

Surveys are very common methods to gather data. There are several survey-based research examples in the GSD field [16,47,48]. The author of this thesis decided to follow a structured methodology for the design, execution and analysis of the surveys based on Barbara Kitchenham's series of articles [35].

The surveys were designed by the author, supervised by the supervisor and validated by the company's responsible to check whether they fitted the interests of both the company and the research. For both surveys, the target population was carefully selected to fulfill the triangulation needed for providing a higher validity to the empirical data.

This section is organized as follows: Section 2.1 presents the design, analysis and results of the developers' survey; section 2.2 presents the design, analysis and results of the product managers' survey; finally, section 2.3 presents joint conclusions for both surveys.

2.1 Developers' survey

This subsection reflects the findings of a survey conducted in the globally distributed company with developers participating from two locations, Sweden and Russia. The purpose of the survey was to identify the state of the current and desired practice on the scale of heavy plan-driven development versus lightweight change-driven (or agile) development from the point of view of developers. Several questions about development and collaboration tools were also added for further research.

Different parts of GSD processes were studied in this survey expecting to help the company in a better understanding of the GSD challenges, and aiming to identify potential problems hindering agility. The rest of this section is organized as follows. Section 2.2.1 presents the survey goals and hypotheses, section 2.1.2 presents the survey questions, variables, piloting of the survey and selected population. Section 2.1.3 presents the retrieved data analysis. Finally, section 2.1.4 presents the summary of the extracted data analysis.

2.1.1 Goals of the Survey and Hypotheses

The aim of this survey was to determine whether the current ways of working are satisfying for the needs of agile practices, specifically validating the hindrances brought by distribution. The researchers did not know much about the software development processes followed by the potential respondents. Therefore, hypotheses were designed for the survey (see Table 2).

	Hypotheses
HD. 1.	***Heavy Traditional*** *- The generated documentation is heavy, i.e. unnecessary waste is being generated.*
	Lightweight agile *- The generated documentation is light weighted, i.e. no unnecessary waste is being generated.*
HD. 2.	***Heavy Traditional*** *- The developers' contact with the Product Managers is not enough to be considered agile.*
	Lightweight Agile *- The developers' contact with the Product Managers is enough to develop software in an efficient way.*
HD. 3.	***Heavy Traditional*** *- The development process is not incremental.*
	Lightweight agile *– The development process is incremental.*
HD. 4.	***Heavy traditional*** *– The development process is not iterative.*
	Lightweight agile *– The development process is iterative.*
HD. 5.	***Heavy Traditional*** *- The progress of the projects is not measured by working software and/or demonstration.*
	Lightweight agile *- The progress of the projects is measured by working software and/or demonstration.*
HD. 6.	***Heavy traditional*** *- Developers are not able to integrate Product Managers' changes in requirements in some stages of the development process.*
	Lightweight agile *- Developers are able to integrate Product Managers' changes in requirements in any stage of the development process.*
HD. 7.	***Heavy traditional*** *- Product Managers' feedback is not always integrated.*
	Lightweight agile *- Product Managers' feedback is always integrated.*
HD. 8.	***Heavy traditional*** *- There is not enough level of intra-team trust and motivation.*
	Lightweight agile *- There is an acceptable level of intra-team trust and motivation.*
HD. 9.	***Heavy traditional*** *– The development teams are not self-organizing*
	Lightweight agile *– The development teams are self-organizing*
HD. 10.	***Heavy Traditional*** *– Requirements are validated by the Product Manager in the late stages of the development process.*
	Lightweight agile *– There is a periodical meeting with the Product Manager, held in each sprint to validate the requirements.*
HD. 11.	***Non efficient GSD environment*** *- Team members are not using the necessary tools to ensure a successful coordination and communication.*
	Efficient GSD environment *- Team members are using the necessary tools to ensure a successful coordination and communication.*
HD. 12.	***Non Efficient GSD environment*** *- Team members do not identify which are the most important tools to ensure a successful Global Software Development.*
	Efficient GSD environment *- Team members identify which are the most important tools to ensure a successful Global Software Development.*
HD. 13.	***Non Efficient GSD environment*** *– Team members do not identify their remote team mates.*
	Efficient GSD environment *– Team members identify their remote team mates*
HD. 14.	***Non Efficient GSD environment*** *- Team members do not know who are their product managers and stakeholders.*
	Efficient GSD environment *- Team members know who are their product managers and stakeholders.*
HD. 15.	***Heavy traditional*** *– Team members only communicate when problems arise.*
	Lightweight agile *– Team members communicate every day in a periodical meeting.*

Table 2. Hypotheses of the Developers' Survey

2.1.2 Variables, Survey Questions, Piloting and Selected Population

According to the goals of the survey, a descriptive survey method [35] was chosen. It was a five-point Likert scale survey (see Table 3) combined with some open questions.

1	*Strongly Disagree*
2	*Disagree*
3	*Undecided/Unsure*
4	*Agree*
5	*Strongly Agree*

Table 3. Legend for the five-point Likert scale survey.

The survey evolved through an iterative design process. The final questionnaire consisted of 24 questions, five open questions and eighteen closed questions based on the five-point Likert scale.

- **Open Questions**: focused on detecting team roles, contact with the Product Manager and the team, and tools used in the daily routine. These questions were posted to also learn about the respondents' location and team.
- **Five point Likert-scale questions**: focused on detecting the current and the desired agility state of the development processes. The lowest values of the variables meant a trend to traditional software development practices while the highest ones meant a trend towards agile practices.

2.1.2.1 Survey Questions

Table 4 shows the survey questions (with the corresponding research questions) by order of appearance. Questions 1, 2, 3, 4 and 5 were open questions. The rest of the questions were 5-point Likert scale closed questions and lists.

ID	Question	Hypotheses
1	Describe who plays the following roles in your project: 1.Manager (s), 2.Stakeholder(s), 3.Product Manager(s)	HD. 12
2	Who are the members of your team?	HD. 12
3	Please, state five tools you consider essential for the success of your project	HD. 11
4	How often do you have face-to-face meetings with the Product Manager(s)?	HD. 10
5	How often do you have face-to-face meetings with your team mates?	HD. 13
6	I work with lightweight documentation as source for developing software	HD. 1
7	I consider that I work with more documentation than needed	HD. 1
8	I have enough contact with the Product Manager(s)	HD. 2
9	I would like to have daily contact with Product Manager(s) to receive continuous feedback	HD. 2
10	I follow an incremental development process	HD. 3
11	I follow an iterative development process	HD. 4
12	My progress is measured by working software	HD. 5
13	I prefer my process to be measured by working software	HD. 5
14	Changes in requirements are allowed, even in the later steps of the development process	HD. 6 and HD. 10
15	I would like to be able to easily integrate requirement changes in any stage of the development process	HD. 6
16	I periodically release or demonstrate useful and working pieces of software to ensure Product Manager satisfaction	HD. 10
17	I feel motivated and trusted by my team mates	HD. 8
18	My team is self-organizing	HD. 9
19	I prefer to work in a self-organizing team than having managers assigning tasks	HD. 9
20	I always integrate Product Managers' feedback in the development process	HD. 6 and HD. 7
21	I am sure that I have the same goals as Product Managers and stakeholders	HD. 10
22	I would like to discuss the alignment of goals with Product Managers and stakeholders	HD. 10
23	I get enough feedback from Product Managers and stakeholders	HD. 2
24	Please, state which of the following tools do you use in your daily working routine	HD. 11

Table 4. Questions of the survey

2.1.2.2 Variables

The variables of the study were defined based on the hypotheses and the survey questions formulated in section 2.1.1 and 2.1.2.1. Next, the defined variables are listed and explained:

- **Current Documentation Weight**: It represents the weight of the documentation the developers currently work with. The data collected in this variable corresponds to the question "I work with lightweight documentation as source for developing software".
- **Desired Documentation Weight**: It represents the weight of the documentation the developers would like to work with. The data included in this variable corresponds to the question "I consider that I work with more documentation than needed".
- **Current Contact with the Product Managers**: It represents the current contact the development team has with the Product Manager. The data included in this variable corresponds to the question "I have enough contact with the Product Manager(s)".
- **Desired Contact with the Product Managers**: It represents the contact the development team would like to have with the Product Manager. The data included in this variable corresponds to the question "I would like to have daily contact with Product Manager(s) to receive continuous feedback".
- **Incremental Process Nature**: This variable detects whether or not the developers are following an incremental development process. The data included in this variable corresponds to the question "I follow an incremental development process".
- **Iterative Process Nature**: This variable detects whether or not the developers are following an iterative development process. The data included in this variable corresponds to the question "I follow an iterative development process".
- **Current Way of Progress Measurement**: This variable shows if the current way of progress measurement is by pieces of useful working software or demonstrations. The data included in this variable corresponds to the question "My progress is measured by working software or demonstrations".
- **Desired Way of Progress Measurement**: This variable shows if the developers are satisfied with their work being measured by pieces of working useful software or demonstrations. The data included in this variable corresponds to the question "I prefer my process to be measured by working software"
- **Current Changes in Requirements Allowance**: This variable represents the flexibility of the development process regarding integration of the Product Managers' feedback and changes in requirements. The data included in this variable corresponds to the question "Changes in requirements are allowed, even in the later steps of the development process"
- **Desired Product Managers' Feedback Integration Easiness**: This variable represents the desired flexibility of the development process in terms of integration of Product Managers' feedback and changes in requirements. The data collected in this variable corresponds to the question "I would like to be able to easily integrate requirement changes in any stage of the development process"
- **Useful Working Software Periodical Releases**: This variable shows if the team ensures Product Managers' satisfaction by releasing pieces of useful software or demonstrations. Data included in this variable corresponds to the question "I periodically release or demonstrate useful and working pieces of software to ensure Product Managers' satisfaction".

- **Team Trust and Motivation**: This variable represents the intra-team motivation and trust. The data collected in this variable corresponds to the question “I feel motivated and trusted by my team mates”
- **Current Team Self-organization**: This variable represents the perception of the developers about the organization of their team. The data collected in this variable corresponds to the question “My team is self organizing”
- **Desired Team Self-organization**: It represents the desired organization way of their team. The data collected in this variable corresponds to the question “I prefer to work in a self-organizing team than having managers assigning tasks”
- **Feedback Integration Frequency**: This variable represents the frequency the feedback from the Product Managers is integrated in the development process. It corresponds to the data extracted from the question “I always integrate Product Managers' feedback in the development process”.
- **Current Requirements Alignment with the Product Manager**: This variable represents the sureness of the developers on requirements alignment with the Product Manager. The data collected in this variable corresponds to the question “I am sure that I have the same goals as Product Managers and stakeholders”.
- **Desired Requirements Alignment with the Product Manager**: This variable represents the desired sureness of the developers on requirements alignment with Product Manager. Data collected in this variable corresponds to the question “I would like to discuss the alignment of goals with Product Managers and stakeholders”.
- **Product Managers’ Feedback Sufficiency**: This variable represents whether or not feedback the developers get from the Product Manager is considered enough. Data collected in this variable corresponds to the question “I get enough feedback from Product Managers and stakeholders”

The agility of each variable is determined based on the Agile Manifesto and/or the company’s PMPAT.

2.1.2.3 Piloting

Once the survey was designed, it was brought under a pilot study as described in [49]. The pilot testing or study had different goals:

- To check whether the questions were understandable.
- To evaluate the reliability and validity of the instrument.
- To ensure that the used data analysis techniques match the expected responses.

A pilot study aimed at identifying missing, unnecessary or ambiguous questions and instructions. The validation was conducted using the same technology and procedure as the original study. The pilot testing group was formed by four developers from four different projects and four different global companies. These developers were contacted via phone or email and, after getting their acceptance to collaborate, they received an email with explanatory instructions and the web based survey attached.

For the analysis of the pilot testing the questions were divided into three main subgroups: quantitative, demographic and qualitative. The division was done depending on the kind of data the questions were expected to retrieve.

The analysis of the quantitative data was performed by dividing the quantitative questions into two main subgroups: one detecting the current state and one detecting the desired one. The subjects belonged to different groups and the main variables measured in this survey had a

question about the current state and the corresponding question about the desired state. An evaluation was done to check whether the questions regarding the desired state were in the direction of improving and adopting agile practices and expanded the reliability retrieved to the questions regarding the current state. The Cronbach test over the collected pilot data retrieved α=0.689. This is an indication of an acceptable level of reliability. No tests were conducted to determine the internal validity, since the small size of the pilot population constrained the possibilities of the pilot study.

For the qualitative and demographic retrieved data a checklist extracted from [50] was run by the author of the thesis. The purpose of this checklist was to detect whether or not the subjects forming the pilot testing had correctly understood the questions and provided useful information. The checklist was formed by the following questions:

- Do the respondents understand the objective of the survey?
- Is the wording of the survey clear?
- Are the answer choices compatible with the respondents' experience in the matter?
- Do the answers collected reflect what we want in regards to the purpose of the survey?
- Is there enough diversity in the answers received?

After running the questions listed above over the pilot data we found two subjects responding to one of the survey qualitative questions in an unexpected manner. In one of the cases, the subject lacked of advanced English knowledge (judged from the grammar and spelling mistakes in the open questions), so the question was misunderstood. In the second case, we detected that the question missed some clarification of what was expected. The question was unclear and incomplete. In response, this and other questions were revisited and explanations were added to ensure the clarity of the survey.

2.1.2.4 Target Population and Execution

The population selected by the company for the study was 27 developers involved in several globally distributed projects. The developers were selected by the responsible from the company. No incentives were offered to complete and submit the lengthy 24 question survey.

The web-based survey link was sent to the representative of the company in December 2010, which was forwarded to the selected subjects with the following instructions:

"The purpose of this survey is to explore the collaboration among the different project members, investigate tools and practices used, and also learn about the preferred approaches to different aspects of project organization. The results of the survey will be used for identifying potential improvements in the project. Participation in the survey is anonymous; its results will be synthesized and made available for the participants."

A survey reminder was sent in January 2011 since the responses at this date were not enough for the study.

2.1.3 Results

In this section the results of the survey variables analysis are presented.

2.1.3.1 Demographical Data Analysis

The survey response rate was 15 (55.5%). The identification of the teams was done by the classification of the different developers responses. Three teams were detected, namely A, B and C (see Figure 4). The distribution of the teams was identified by analyzing the replies to the questions related to roles. With this information, the developers and managers could be grouped into teams. Because the survey was organized with no prior information, social networks

(Facebook [51] and LinkedIn [52]) were used to locate the team members geographically. Four developers were impossible to locate in any of these three teams due to inconsistencies or incompleteness in their replies. The resulting configuration of teams is as follows:

- **Project A** is formed by two developers in Russia and four developers in Sweden. There are three managers for this team, one in Russia and two in Sweden.
- **Project B** is formed by four developers in Russia. There are three managers for this team, one located in Russia and two in Sweden.
- **Project C** is formed by five developers, four in Russia and one in Sweden. The manager of this group is located in Sweden.

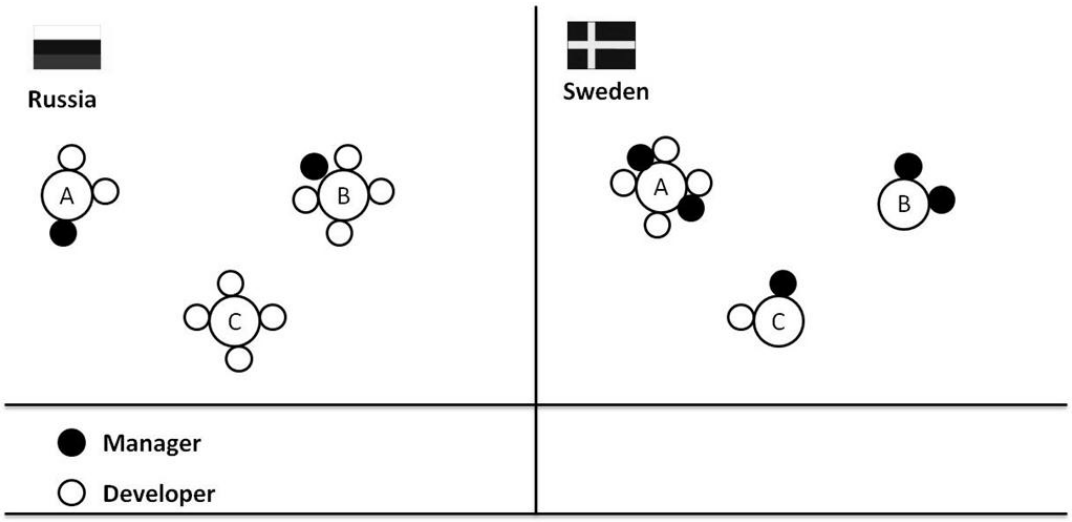

Figure 4. Configuration of the Teams.

This analysis gave the perception the developers have about the team. All of the replies regarding the team mates on the developer level matched with each other inside the same team. As a result, they have a consistent view about who their team mates are, which is beneficial for a successful GSD environment.

The Unit manager (UM), Product Manager (PdM) and Stakeholder roles were detected to be confused inside the development teams. Some developers replied that they do not really know who the PdMs and Stakeholders referring to a whole department. Differences in replies regarding these roles have been detected among the members from Team B. As a result roles seem not to be well defined. This is not beneficial for a GSD team. Further investigation was done regarding this issue.

2.1.3.2 Quantitative data

The quantitative data was studied through scatter plots analysis. The small respondent's size did not allow running any non parametric test (U Mann-Withney, Kolmorogov Smirnov and Pearson's Chi Square) over the retrieved data.

The data from most of the variables is not analyzed in teams but in scatter plots. The X axis represents the 5 points of the Lickert scale and the Y axis represents the number of respondents. This is performed this way to analyze the data from the role perspective and not from the team perspective. It will show the distribution of the responses allowing (when possible) to draw conclusions. As a result, the respondent's size for analysis is 15.

2.1.3.2.1 Current vs. Desired Documentation Weight

Figure 5 shows the scatter plots for the 15 values on Current and Desired Documentation Weight.

- **Current Documentation Weight:** Two developers totally agree on being working with lightweight documentation and two disagree or strongly disagree. The respondents are mostly concentrated on agree and undecided. We can conclude that more than 53.3% of the developers think to be working with lightweight documentation. 33.3% give an undecided answer and 13.3% disagree or strongly disagree with the statement.
- **Desired Documentation Weight:** 33.3% of the respondents report an undecided answer. Only one of the respondents considers being working with more documentation than needed. As a result, 60% of the respondents think that they do not work with more documentation than needed.

In summary, more than 50% of the developers reported to be working with lightweight documentation in the Current Documentation Weight variable and more than 50% of the respondents reported not to be working with more documentation than needed in the Desired Documentation Weight Variable. However, there is a big set of respondents that reported an undecided answer in both variables. Further research in these variables was conducted.

The documentation is perceived as medium-weight, i.e. half agile. However, the developers did not express a desire to change the documentation weight.

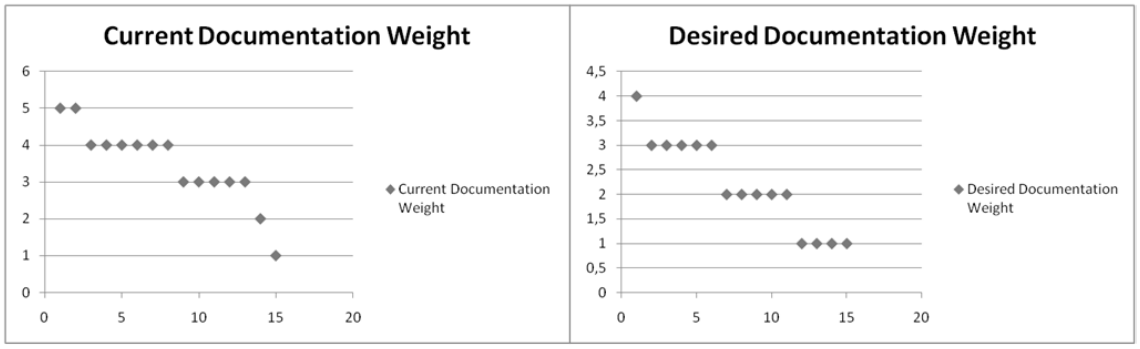

Figure 5. Current vs. Desired Documentation Weight

2.1.3.2.2 Current vs. desired contact with the Product Manager

Figure 6 shows the box plots for the 15 values on Current and Desired Contact with the Product Manager.

- **Current Contact with the Product Manager:** 53.3% of the respondents are concentrated between agree and totally agree on having enough contact with the Product Manager. Three developers give an undecided answer and three disagree or strongly disagree.
- **Desired Contact with the Product Manager:** 60% of the respondents give an undecided answer on the desire of maintaining daily contact with the Product Manager. They seem unsure about the possible benefits of these meetings. Only three subjects agree or absolutely agree on maintaining a daily contact with the Product Manager. Four of them disagree or absolutely disagree.

In summary, more than 50% of the respondents consider the contact with the Product Manager sufficient and more than 50% of the respondents would not like to have a daily meeting with the Product Manager. This is quite good considering the distribution of the teams

and the PMs. The subjects reporting not having enough contact with the Product Manager and wishing to have a daily meeting with him/her are the same respondents.

In contrast with the survey responses, the PMPAT clearly states that all team members (including the PdM) should have a daily meeting. As a result, the daily meetings are not generally conducted. The current state of agility is acceptable, but there is still room for improvement.

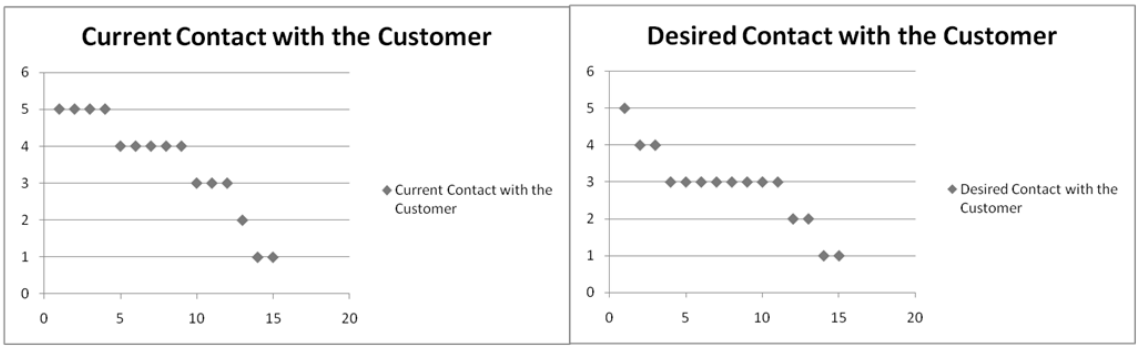

Figure 6. Current vs. Desired Contact with the Product Manager

2.1.3.2.3 Incremental and Iterative process

Figure 7 shows the scatter plots for the 15 values on Incremental and Iterative Process Nature.

- **Incremental Process Nature**: The responses on this variable are very disperse. Five respondents consider their process incremental and seven disagree or strongly disagree with this statement. It shall be noted that three respondents give an undecided answer. The undecided answers may indicate a blurred understanding of the development process.
- **Iterative Process Nature**: 60% of the respondents consider their development process iterative and two strongly disagree with this consideration. Four developers give an undecided answer. The development processes are mostly iterative, but the four undecided answers may indicate a blurred understanding of the development process nature.

Agile practices and the PMPAT recommend following an incremental and iterative development process. The rate of the variable that detects the perceived view of the development process being incremental is less than 50%. The development processes are reported to be mostly iterative. The level of agility in the process nature is improvable. The undecided responses indicate that the development process nature is blurry for some developers. Further investigation has been carried out in the software development process nature and it is later depicted.

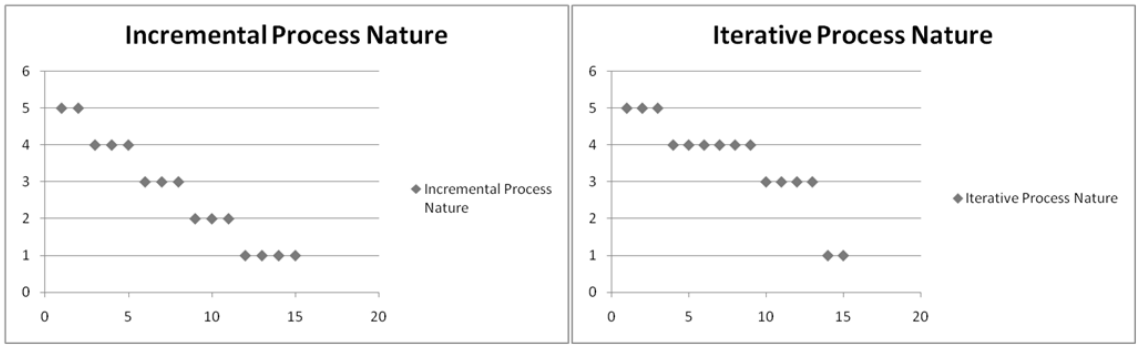

Figure 7. Incremental and Iterative Process Nature

2.1.3.2.4 Current vs. Desired Way of Progress Measurement and Useful Software Periodical Releases

Figure 7 shows the scatter plots for the 15 values on Current and Desired Way of Progress Measurement.

- **Current Way of Progress Measurement:** 80% of the respondents reported that their progress is measured by working software or demonstrations. 25% could not decide.
- **Desired Way of Progress Measurement:** A vast majority of the respondents reported a preference for their progress to be measured by working software or demonstrations. Three respondents reported an undecided answer and one reported disagreement. This last one had responded an undecided answer in the Current Way of Progress Measurement variable.
- **Useful Software Periodical Releases:** 40% of the respondents have reported to be releasing pieces of useful software to ensure Product Manager's satisfaction. This percentage of the respondents is concentrated in the "agree" point. 26.6% disagree or strongly disagree with this statement.

Both the PMPAT and the Agile Manifesto recommend to measure software progress by pieces of working software. The most common way of progress measurement among respondents is by working software or demonstrations. The developers mostly reported to prefer this measurement, which suggests a high level of agility in this variable.

In the PMPAT it is stated that at the end of the development phase the results should be shown to the PM in form of demonstrations. However, less than 50% of the respondents reported to be releasing **periodically** useful pieces of software for demonstrations. The way of measuring the progress is by demonstrations or pieces of working software, but it is not detected to be done in a periodical way.

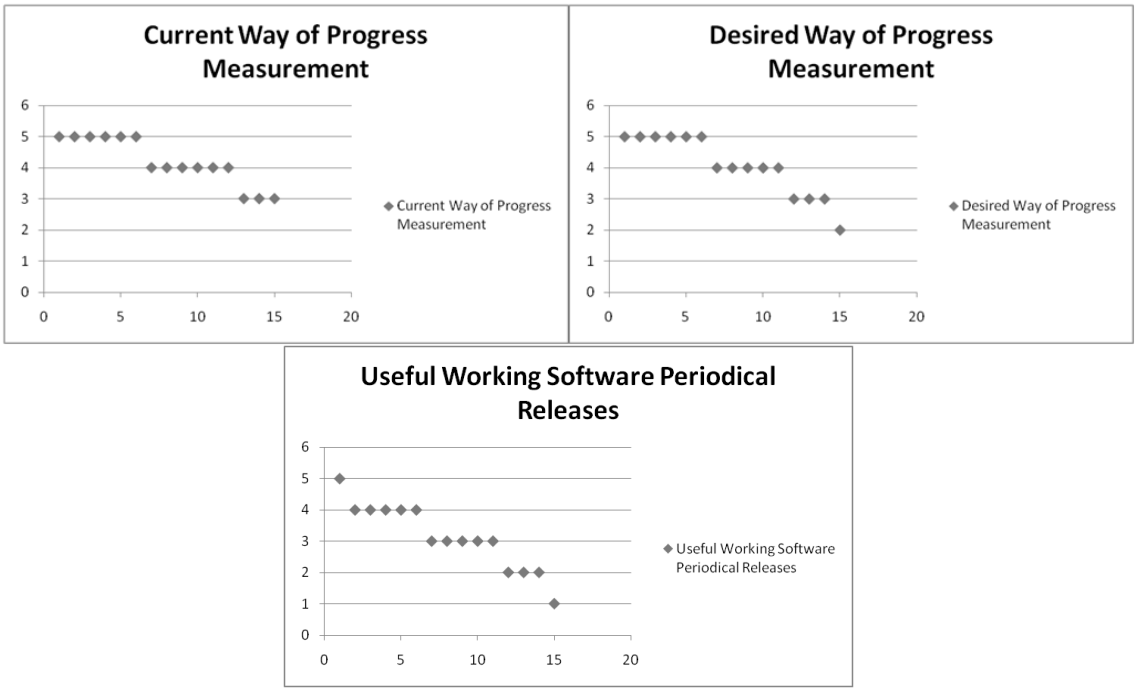

Figure 8. Current vs. Desired Way of Progress Measurement and Useful working Software Periodical Releases

2.1.3.2.5 Current Changes in Requirements Allowance and Feedback Integration Frequency

Figure 9 shows the scatter plots for the 15 values on Current Product Managers' Feedback Integration Allowance and Frequency.

- **Current Changes in Requirements Allowance:** The respondents' responses are very spread in this variable. 46.6% of the respondents reported agreement on currently allowing changes in requirements in any stage of the development process while 33.3% disagree or strongly disagree on it. The negative responses come from subjects belonging to three different teams, so this perception is spread in all the products.
- **Feedback Integration Frequency:** More than 50% respondents reported that they always integrate Product Managers' feedback while three reported not always integrating it. Two were undecided.

There are several chances identified in the PMPAT when the Project Manager can include changes in the product backlog taken from the PdM's feedback, one in the planning of the iteration, another one in the revision of the product backlog and another one in the results demonstration. However, even if the PdM can suggest changes in different stages of the development process, some developers reported not being able to implement them at all stages of the development process. On the frequency of integration, more than 66.6% of the respondents reported to always integrate Product Managers' feedback.

There seems to be projects in which the Current Changes in Requirements Allowance variable is more agile than others. The Feedback Integration Frequency looks quite agile, with some exceptions. Hence, there is room for improvement in both variables, but especially in the first one. Further investigation was carried out about this and it is reported in later chapters of the thesis.

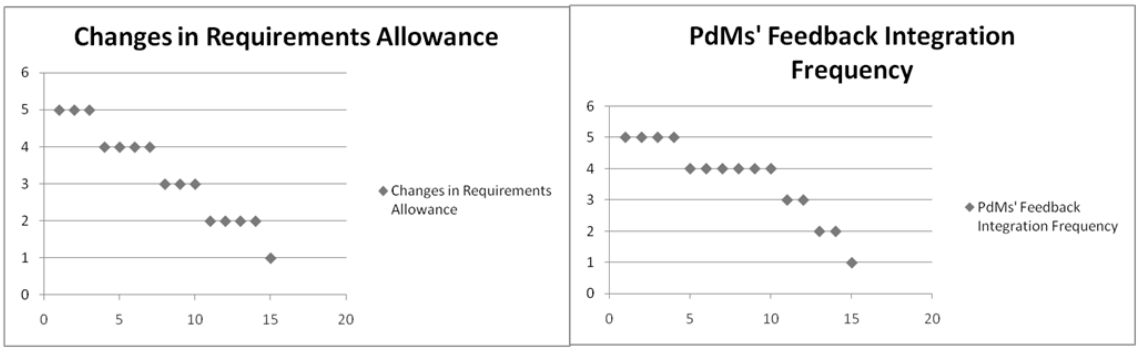

Figure 9. Product Managers' Feedback Integration Allowance and Frequency

2.1.3.2.6 Product Managers' Feedback Sufficiency and Desired Product Managers' Feedback Integration Easiness

Figure 10 shows the scatter plots for the 15 values on Current Product Managers' Feedback Integration Allowance and Completeness.

- **Product Manager's Feedback Sufficiency**: Less than 26.6% of the respondents consider the feedback from the Product Managers sufficient. Six developers give an undecided answer while more than 33.3% disagree or totally disagree with the sufficiency of Product Managers' feedback.
- **Desired Product Managers' Feedback Integration Easiness**: More than 73.3% of the developers would like to integrate Product Managers' feedback in an easy way in any stage of the development process. Just one would not like to do it.

As stated in 2.1.3.2.5, the PdMs have several chances to include report feedback and changes in requirements. The PdM also has to accept the backlog before the developers start coding. However, the developers do not consider the received feedback as sufficient and would like to easily integrate PdM' feedback in any stage of the development process. The proactivity of the PdMs seems not to be agile enough in some cases which could be caused by the distribution of teams and PdMs. The developers would like to be more agile integrating the PdMs feedback even if they seem quite on the line of a good agility.

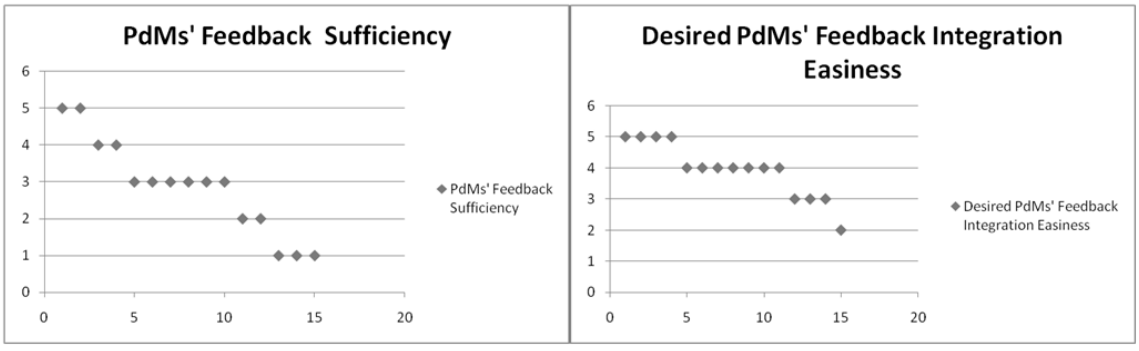

Figure 10. Product Managers' Feedback Sufficiency and Desired Product Managers' Feedback Integration Easiness

2.1.3.2.7 Trust and motivation

Figure 11 shows the scatter plot for the 15 values on Trust and Motivation.

All of the respondents except one are concentrated between agree and strongly agree. The developers feel trusted and motivated. This is essential for a distributed agile development since

one of the agile principles recommend to *"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."*[41]

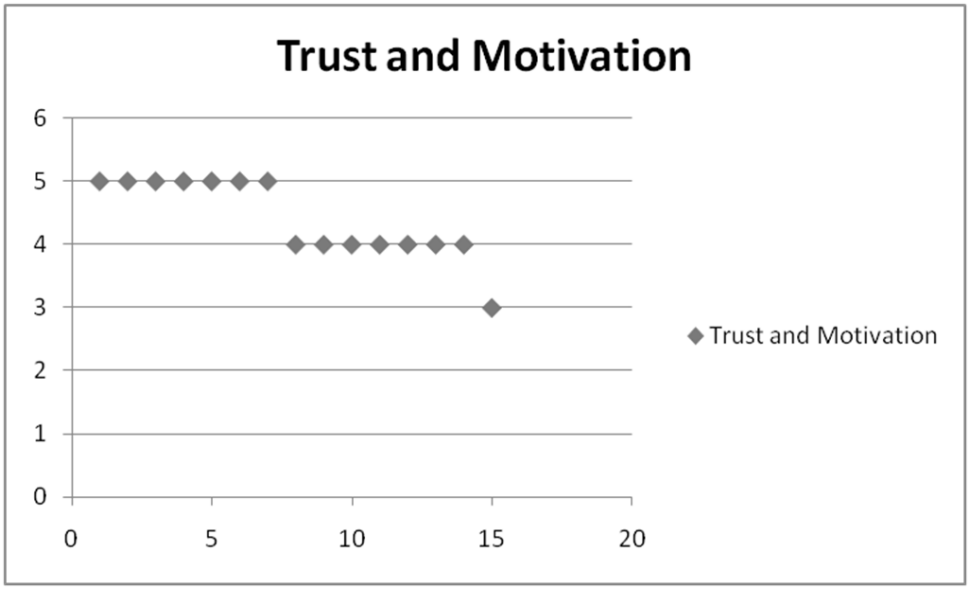

Figure 11. Trust and Motivation

2.1.3.2.8 Current vs. Desired Team Self-organization

Figure 12 shows the scatter plot for the 15 values on Current and Desired Team Self-Organization.

- **Current Team Self-Organization:** 66.6% of the respondents reported that their teams are self-organizing. Five developers disagree or strongly disagree. Three give an undecided answer, what could indicate blurriness about the organizing way of their teams or lack of familiarity with the terminology or the concept of self-organization.
- **Desired Team Self-organization**: 60% of the developers reported preferring to be self-organizing but three of them prefer the opposite. Three developers report an undecided answer.

In summary, the majority of the developers consider their teams self-organizing. However, the number of developers that report their teams to be self-organizing is lower than the ones that prefer to be self-organizing. The developers reporting disagreement with the first variable also reported disagreement with the second one. This could be caused by a resistive attitude towards a change, a lack of knowledge of how a self-organizing team works or a resistance to take more responsibility. Agile recommends teams to be self-organizing [41]. We can conclude that the agility level is still improvable, but the majority of developers feel satisfied with the organizing way.

Figure 12. Current vs. Desired Team Self-organization

2.1.3.2.9 Current vs. Desired Requirements Alignment with the Product Manager

Figure 13 shows the scatter plot for the 15 values on Current vs. Desired Requirements Alignment with the Product Manager.

- **Current Requirements Alignment with the Product Manager:** The majority of the developers reported to be sure about the alignment of requirements with the Product Manager. Two developers replied "absolutely disagree", which means that they are not sure about the requirements alignment.
- **Desired Requirements Alignment with the Product Manager:** The majority of the respondents would like to discuss the requirements alignment with the Product Manager. Only one subject disagrees with this statement.

In summary, 66.6% of the respondents reported to be sure about the alignment of requirements with the Product Manager. 60% of the respondents would like to discuss this alignment with the Product Manager. The developers consider being agile in the alignment, but they would like to be more.

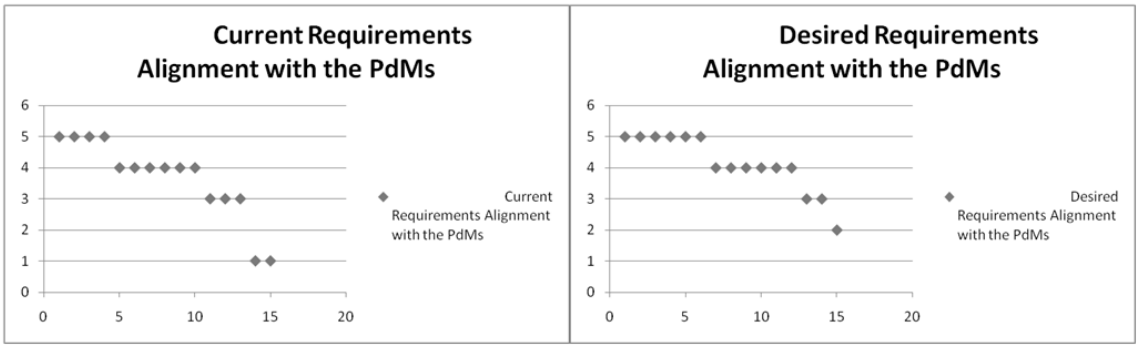

Figure 13. Current vs. Desired Requirements Alignment with the Product Manager

2.1.3.2.10 Product Manager Meetings[2]

Figure 14 shows the scatter plot for the 15 values on Product Managers Meetings.

The majority of the replies are concentrated between 1(Never) and 2(Almost Never). Only two developers reported Weekly frequency of the meetings and three developers reported Daily meetings. In agile software development it is recommended to have daily face-to-face meetings

[2] Data for Product Managers Meetings and Intra-Team Meetings variables was retrieved through open questions. Thus, the gathered data was qualitative. It has been transformed to quantitative with the following scale: 1- Never, 2- Almost Never, 3- Monthly, 4 – Weekly and 5 – Daily.

with the product manager or internal Product Manager since *"Business people and developers must work together daily throughout the project"* [41].. In fact, the PMPAT states that the whole team should participate in a daily meeting to describe which tasks have been completed since the last daily meeting, and what is blocking their progress.

Daily face-to-face meetings are the ideal way of coordinating and communicating in GSD environment. However, they are difficult to be accomplished. A weekly videoconference could be a way to compensate the daily face-to-face meetings.

We can conclude that major improvements are advised in this variable. These improvements can positively affect the agility of other variables (Current and Desired Contact with the PdM, PdM Feedback Integration Allowance, PdM Feedback Integration Frequency, PdM Feedback Sufficiency, PdM Feedback Integration Easiness, Current and Desired Requirements Alignment with the PdM)

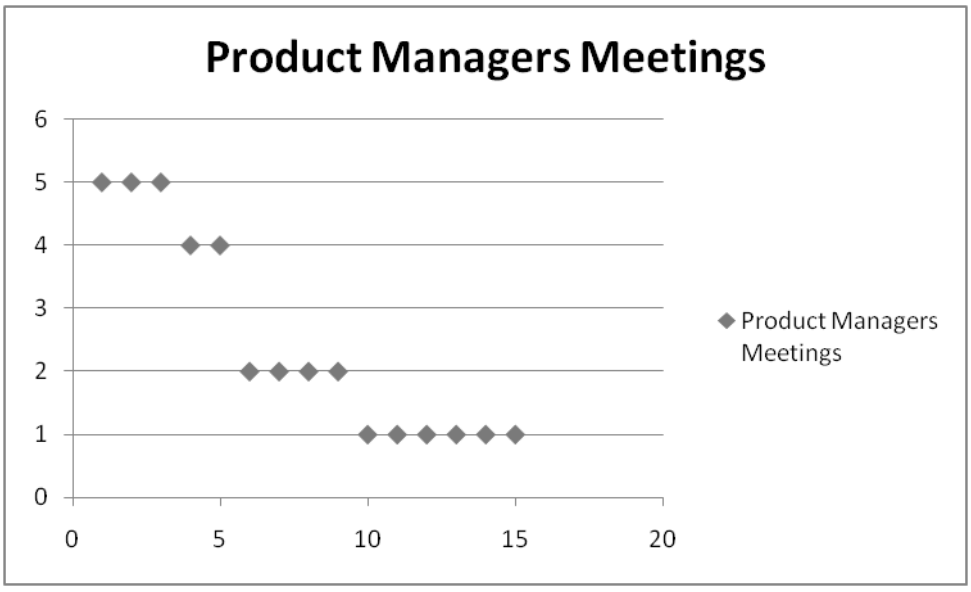

Figure 14. Product Managers Meetings

2.1.3.2.11 Intra-Team Meetings

Figure 15 shows the scatter plot for the 15 values on Intra-Team Meetings.

The PMPAT states that the whole team should participate in a daily meeting, however only 53.3% of the respondents reported to have daily or weekly meetings with all the development team. 33.3% of the respondents never or almost never have face-to-face meetings with the whole development team. This last lack of meetings could be caused by the distribution of the teams. The high level of distribution in this variable appears because some respondents belong to different distributed development teams. Improvements in this variable could affect positively the agility of the development process since it resides in the agility of the development team.

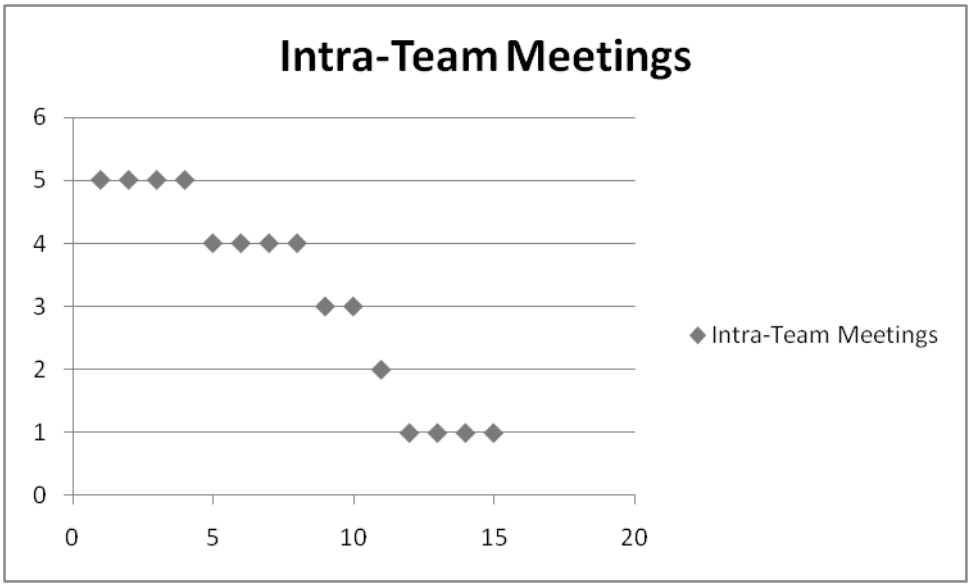

Figure 15. Intra-Team Meetings

2.1.3.3 Qualitative Data Analysis: Tools

GSD is usually affected by geographical, socio-cultural and/or temporal distance. These distances provoke different challenges in communication, coordination and control [42]. Nevertheless, collaboration tools can be used to alleviate these difficulties. The data analyzed in this section comes from two different questions, one related to the tools used in a daily routine and the other to the 5 essential tools for the success of the project the developers are involved in.

Figure 16 shows the tools used in daily routine vs. the tools considered important for the development success. The tools used more are Version Control Systems and email. This means that the main verbal communication point is email. The next communication tools used are Chat and Telephone. They are still less personal, and thus not suitable for increasing teamness and trust. Video-conferences and audio-conferences are used by less than half of the respondents in the daily routine and only two developers use social networks.

The results about the tools considered important contrast with the daily used ones. In the case of the Version Control Systems the response rate is maintained. The importance that the respondents give to a proper and good Code Development Environment and a Version Control System remarkably contrasted with communication tools. When it comes to interaction among the developers, the respondents have different preferences — some choose chat, others would prioritize video-conferences. It is surprising that only one developer prioritize video-conferences as essential, while this collaborative technology is regarded as the one providing richest interaction possibilities.

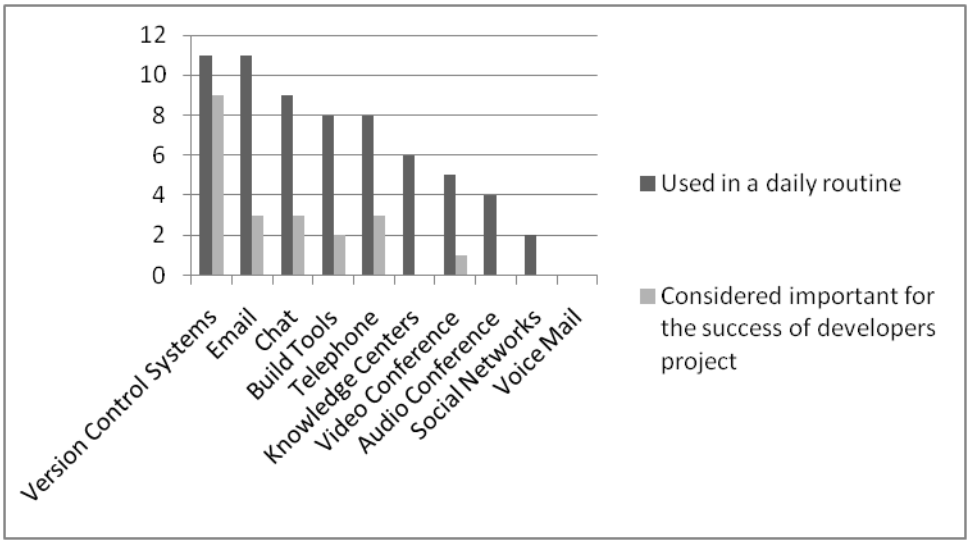

Figure 16. Daily used tools vs. necessary tools

One developer has pointed out some complaints regarding software tools:

- Current document storage (server) does not work as expected.
- A fully equipped test lab is demanded but not available.

Even if the developers have available the necessary computer mediated collaborative tools, the tools do not work as they expect or they would like them to work. This can constrain the agility of the developers. It is necessary to have collaborative tools to alleviate geographical, temporal and socio-cultural distance, but it is also necessary to make them agile. This information was reported to the representative of the company so that they could improve this issue.

2.1.4 Summary

This section summarized the findings of the survey conducted in the company involving developers from Sweden and Russia. The resulting data analysis detected very interesting findings that leave some place for improvements:

- Roles involved in the development process seem not to be well understood by the respondents.
- The documentation weight is heavy to be agile. However, the developers do not express a strong desire of change the documentation weight.
- The current state of agility in contact with the PM is accepted by the developers, but it is improvable from the point of view of the PMPAT.
- The process agility is not reported to be pure iterative or incremental. Thus, it could be more agile.
- The way of measuring progress is quite agile.
- The Product Managers' Feedback Integration Ability variable agility is improvable. The Feedback Integration Frequency is mostly agile.
- The agility level on team self-organization is improvable, but the developers feel satisfied with the current state.
- The respondents would like to be more agile in requirements alignment with the Product Manager.

- The meetings between the PdM and the team are not being accomplished with the PMPAT specification. This could be affecting the agility of the other variables.
- Two of the development teams were distributed between Russia and Sweden and the other one was co-located in Russia. Product Management in all three projects was performed on the distance from Sweden. Despite this difference, complete intra-team trust and motivation were detected in all developers. This is important for a successful Agile Distributed Development.
- The main used daily communication tool is email, followed by phone and chat. Video conference and social network tools are among the least mentioned.
- The tools considered essential for the success of the developers' projects are version control systems, code development environments and office package (in respective order).

In conclusion there seem to be agile features combined with traditional development. The developers express an acceptance to agile practices. A more spread use of collaborative face-to-face tools providing interaction among remote team mates (video-conferences, video-messages, etc.) is advised due to the distribution of the teams. They are not detected to be commonly used in the developers' daily routine.

Development process clarification, customers' feedback integration and roles misunderstanding were taken as main issues to be addressed in next steps of the research methodology.

2.2 Product Managers' survey

This subsection reflects the findings of a survey conducted in the company. The purpose of the survey was to determine the degree of agility the product managers had and if they were satisfied with it. Some questions were also added to clarify whether or not the product managers can perform their role in a prescribed way, in particular, whether the distribution forces them to behave as the anti-product managers. This means, that e.g. the product manager is not able to attend the demo, is unable to attend to the sprint planning, etc. So, through the questionnaire we aim to detect potential problems the product managers have in their interaction with the development team.

The rest of the section is organized as follows: Section 2.2.1 presents the survey goals and hypotheses. Section 2.2.2 presents the survey questions, variables, piloting of the survey and selected population. Section 2.2.3 presents t the analyzed results. Finally, Section 2.2.4 presents a summary of the analyzed data.

2.2.1 Goals of the survey and Hypotheses

The survey was designed by David Musat Salvador and Darja Šmite following Barbara Kitchenman's guidelines [35]. The aim of the survey was to determine the degree of agility of the product managers and their satisfaction with it, detecting problems in the role behavior caused by the distribution. Based on the results obtained from the developers' survey some hypotheses were formulated (see Table 5).

Hypotheses	
HPM. 1.	***Heavy Traditional:*** *The communication protocols are not suitable for agile practices, i.e. not frequent face-to-face communications and/or videoconferences.*
	Lightweight Agile: *The communication protocols are suitable for agile practices, i.e. frequent face-to-face communications and/or videoconferences.*
HPM. 2.	***Non Efficient GSD Environment:*** *The PdMs do not consider communication, coordination and control tools necessary for the success of the products they manage.*
	Efficient GSD Environment: *The PdMs consider communication, coordination and control tools necessary for the success of the products they manage.*
HPM. 3.	***Heavy Traditional:*** *There is no periodical frequent meeting between developers and PdMs.*
	Lightweight Agile: *The PdMs have frequent meetings with the development team.*
HPM. 4.	***Heavy Traditional:*** *The development process generates more documentation than needed.*
	Lightweight Agile: *The development process does not generate more documentation than needed.*
HPM. 5.	***Heavy Traditional:*** *Feedback from the PdMs is not fully integrated in the product.*
	Lightweight Agile: *Feedback from the PdMs is fully integrated in the product.*
HPM. 6.	***Heavy Traditional:*** *The communication process between PdMs and developers is slow and not effective.*
	Lightweight Agile: *The communication process between PdMs and developers is quick and effective.*
HPM. 7.	***Non Efficient GSD Environment:*** *The PdMs are not able to understand and/or identify the development process nature.*
	Efficient GSD Environment: *The PdMs understand and identify the development process nature.*
HPM. 8.	***Heavy Traditional:*** *The PdMs cannot measure the product progress by pieces of working software or demonstrations.*
	Lightweight Agile: *The PdMs can measure the product progress by pieces of working software or demonstrations.*
HPM. 9.	***Heavy Traditional:*** *Changes in requirements are not welcomed by the development team in some stages of the development process.*
	Lightweight Agile: *Changes in requirements are welcomed by the development team, even in late stages of the development process.*
HPM. 10.	***Heavy Traditional:*** *Changes in requirements are dramatic between iterations.*
	Lightweight Agile: *Changes in requirements are constant between iterations, but not dramatic.*
HPM. 11.	***Non Efficient GSD Environment:*** *The PdMs do not feel trusted by the development team.*
	Efficient GSD Environment: *The PdMs feel trusted by the development team.*
HPM. 12.	***Heavy Traditional:*** *The PdMs are not able to play their role correctly behaving as anti-PdMs.*
	Lightweight Agile: *The PdMs do not have to behave as anti-PdMs.*
HPM. 13.	***Non Efficient GSD Environment:*** *The PdMs do not consider communication, coordination and control tools essential for the success of the development of their products.*
	Efficient GSD Environment: *The PdMs consider communication, coordination and control tools essential for the success of the development of their products.*
HPM. 14.	***Heavy Traditional:*** *Deadlines are not successfully achieved.*

	Lightweight Agile: *Deadlines are successfully achieved.*
HPM. 15.	***Heavy Traditional:*** *The PdMs are not satisfied with the way they play their role.*
	Lightweight Agile: *The PdMs are satisfied with the way they play their role.*
HPM. 16.	***Non Efficient GSD Environment:*** *The PdMs do not know and/or not understand the development process.*
	Efficient GSD Environment: *The PdMs know and understand the development process.*
HPM. 17.	***Non Efficient GSD Environment:*** *The PdMs do not use suitable collaborative tools in their daily routine.*
	Efficient GSD Environment: *The PdMs use suitable collaborative tools in their daily routine.*
HPM. 18.	***Non Efficient GSD Environment:*** *The PdMs do not correctly recognize the roles involved in the development process*
	Efficient GSD Environment: *The PdM correctly recognize the roles involed in the development process.*
HPM. 19.	***Heavy Traditional:*** *The PdMs do not identify their responsibilities.*
	Lightweight Agile: *The PdMs know their responsibilities and identify them*

Table 5. Hypotheses for the PdMs survey.

2.2.2 Variables, Survey Questions, Piloting and Selected Population

The survey was designed to gain a deep understanding on the current and desired agility of the PdMs and problems caused by the distribution. This section describes the background and the survey study design.

According to the goals of the study, a descriptive survey method was chosen [35]. It was a five-point Likert scale survey (see Table 6) combined with some open questions.

1	*Strongly Disagree*
2	*Disagree*
3	*Undecided*
4	*Agree*
5	*Strongly Agree*

Table 6. Legend for the five-point Likert scale survey.

The survey evolved through an iterative design process. The questionnaire was formed by 32 questions: four open questions, one multi-choice question, two combo box questions, three list questions and twenty two closed questions, the last ones in form of 5-point-Likert scale.

- **Open Questions**: They were mainly focused on detecting the communication protocols between developers and PdMs, used tools to overcome the distribution, and exploring the reasons of changes in requirements in the development process. There was also one open question to express suggestions about improvements in the development process.
- **Five Point Likert-scale Questions**: They were designed to detect the current agility, the desired one and to capture the PdMs concept of the development process
- **Multi-choice Question:** It was designed to detect the frequency the PdMs have meetings with the development team to coordinate through the distribution.
- **Combo box questions:** They were designed to detect the concept the PdMs have about the development process nature.
- **List questions.** They were mainly focused on detecting if there was any confusion on the role the PdMs were playing caused by the distribution, which roles the PdMs could identify inside the development process and to collect the main tools a PdM uses in a daily routine to overcome the geographical distance.

2.2.2.1 Survey Questions

Table 7 shows the survey questions (with the corresponding research questions) by order of appearance. Questions 1, 2 and 4 were open questions. 27 and 28 were combo box questions. Questions 29 and 30 were list questions. Question 3 was a multi-choice question. The rest of the questions were 5-point Likert scale closed questions. There was also an open question for the respondent to express improvements in the development process: *"If you could change something from the development process you are currently following, what would you change?"* Data from two questions was not included in the study because it lacked relevance.

ID	Question	Hypotheses
1	Please, describe the communication process with the development team	HPM. 1
2	Please, state five tools you consider essential for the success of a distributed project	HPM. 2, HPM. 13
3	How often do you have face-to-face-meetings with your team?	HPM. 1, HPM. 3
4	Please describe the main reasons of changes in requirements during the development process	HPM. 10
5	I consider that the development process generates more documentation than needed	HPM. 4
6	I consider that the feedback I provide is not fully integrated in the final product	HPM. 5
7	Communication with the development team is slower than I would like to	HPM. 1, HPM. 6
8	Products are built in an incremental way	HPM. 7, HPM. 16
9	Products are built in an iterative way	HPM. 7, HPM. 16
10	I can measure the progress of the product by working software or demonstrations	HPM. 8
11	I prefer to measure the progress of the product by working software or demonstrations	HPM. 8
12	Developers welcome changes in requirements, even in the later steps of the development process	HPM. 9
13	Requirements change dramatically from one iteration to the following one	HPM. 10
14	It is very common to have changes in requirements in later steps of the development process	HPM. 10
15	I feel trusted by the development team	HPM. 11
16	I am able to attend to every meeting in which software pieces are demonstrated	HPM. 12
17	I am able to coordinate with the development team a prioritized list of requirements every time we have a meeting	HPM. 12
18	I would like to take into account the development team to established a prioritized list of requirements	HPM. 12
20	I am able to provide support when a development team needs me	HPM. 12
22	I find it difficult to coordinate dates and appointments to perform my role in an optimal way	HPM. 12
23	I have clear understanding of the development process structure	HPM. 7, HPM. 16
24	I would like to have more explanatory and defined development process definition	HPM. 7, HPM. 16
25	The development process is well defined and explained	HPM. 7, HPM. 16
26	The development process is well understood	HPM. 7, HPM. 16
27	A software product is split in different features	HPM. 7, HPM. 16
28	One feature is assigned to one team	HPM. 16
29	Please, state which of the following tools you use in your daily working routine	HPM. 17
30	Please, state which of the following roles you recognize in your project	HPM. 18
31	Please, state which of the following roles you play in your project	HPM. 19

Table 7. Questions of the Product Managers Survey

2.2.2.2 Variables

The variables of the study were defined based on the hypotheses and the survey questions formulated in sections 2.1.1 and 2.2.2.1. Next, the defined variables are listed and explained:

- **Current Documentation Overproduction**: It represents the potential extra unnecessary documentation generated in the development process. The data collected in this variable corresponds to the question "I consider that the development process generates more documentation than needed".
- **Current Degree of PdM Feedback Integration**: It represents the PdM's perception about the grade in which their feedback is integrated in the final product. The data included in this variable corresponds to the question "I consider that the feedback I provide is not fully integrated in the final product".
- **Communication Speed Satisfaction**: It represents the satisfaction the PdMs have with the communication process speed between PdMs and developers. The data included in this variable corresponds to the question "Communication with the developers team is slower than I would like to".
- **PdMs Process Nature Perception**: It represents the PdMs' perception of the nature of the development process their product is following. The data included in this variable corresponds to the questions "Products are built in an incremental way", "Products are built in an iterative way", "The product is spilt in features" and "One feature is assigned to one team".
- **Current Progress Measurement Way**: This variable evaluates the grade in which the current way of progress measurement is done by pieces of working software or demonstrations. The data included in this variable corresponds to the question "I can measure the progress of the product by working software or demonstrations".
- **Desired Progress Measurement Way**: This variable evaluates the grade the PdMs would like to evaluate the progress of a product by working software or demonstrations. The data included in this variable corresponds to the question "I prefer to measure the progress of the product by working software or demonstrations".
- **Changes in Requirements:** This variable represents the way requirements change during the development process, how they are accepted by the development team and why these changes are produced. The data included in this variable corresponds to the questions "Developers welcome changes in the requirements, even in the later steps of the development process", "Requirements change dramatically between iterations" and "It is very common to have changes in requirements in later steps of the development process".
- **Trust**: This variable represents the trust the PdMs perceive from the development team. The data collected in this variable corresponds to the question "I feel trusted by the development team".
- **Capability to Play the PdM Role**: This variable represents the capability the PdMs have to play the PdM role in a proper way, i.e. not to be forced to behave as an anti PdM. The data collected in this variable corresponds to the questions "I am able to attend to every meeting in which software pieces are demonstrated", "I am able to coordinate with the development team (without any problem) a prioritized list of requirements every time we have a meeting", and "I am able to provide support when a development team needs me".

- **Willingness to Prioritize Requirements:** This variable represents the willingness of the PdMs to prioritize requirements with the development team. The data collected in this variable corresponds to the question "I would like to take into account the development team to establish a prioritized list of requirements".
- **Current Development Process Understanding:** This variable represents the way PdMs understand the current development process. The data collected in this variable corresponds to the data extracted from the question "I have clear understanding of the development process structure", "The process development is well defined and explained" and "The process development is well understood".
- **Desired Development Process Clarity:** This variable represents the PdMs desired understanding of the development process. The data collected in this variable corresponds to the data extracted from the question "I would like to have a more explanatory and defined development process definition".

2.2.2.3 Piloting

Once the survey was designed, it was brought under a pilot study as described in section 2.1.2.3. The pilot study population consisted of two product managers from two distributed companies. They were involved in different distributed projects.

For the analysis of the piloting the questions were divided in two main subgroups: quantitative and qualitative (demographic questions were not included in the first version of the survey). This division was done depending on the kind of data the questions were supposed to retrieve. The Cronbach test could not be run over the pilot survey since the pilot population was very small to provide conclusive α. However, data was observed and, excepting the case of the Documentation Overproduction variable, the rest of the answers varied one point up or down from each other. This meant an acceptable reliability level.

For the qualitative and demographic questions the same procedure as in section 2.1.2.3 was followed. After running the validation over the qualitative data a missing answer corresponding to the question "If you could change something from the development process you are currently following, what would you change?" was detected. This could be caused by several reasons: 1. the subject totally agreed with how the development process was working or 2. the subject was resistive to reveal any kind of intra-company information. The rest of the questions were well understood and correctly answered. They provided interesting and useful information for the study.

Additional evaluation was requested from the IBM Rational software process responsible in Spain. The reported recommendations were:

- *Reduce the open questions. People feel lazy to write long sentences and statements. There are usually lots of resulting non valid answers.*
- *Translate the open questions asking for frequency, i.e. "How often…?" to Likert scale in the way 1= Never…5=Always.*
- *Avoid questions containing sensitive information regarding names and confidential information.*

Changes in the survey were introduced following the provided suggestions and the findings of the pilot study. Open questions were reduced and some of them were translated to combo lists.

To avoid the problems in further analysis, two questions asking respondent's name and the product name were added in the final version. These two last completed the survey design forming the demographical data set.

2.2.2.4 Selected Population

The population selected by the company for the study was formed by eight PdMs involved in several projects. The PdMs were selected by the representative from the company in charge of the project. No incentives were offered to complete and submit the lengthy 32 question survey.

The web-based survey link was sent to the representative of the company at the beginning of 03/2011 which was forwarded to the selected subjects with the following instructions:

"The purpose of this survey is to explore the collaboration among the different project members, investigate tools and practices used, and also learn about the preferred approaches to different aspects of project organization. The results of the survey will be used for identifying potential improvements in the project. Participation in the survey is voluntary; its results will be synthesized and made available for the participants."

A reminder of the survey was sent at the end of 03/2011 since there was only one respondent upon that date.

2.2.3 Results

In this section, the results of the survey are analyzed. The survey had a response rate of four (50%). The gathered data provided interesting findings on tools, software process models and PdMs agility.

2.2.3.1 Demographical Data Analysis

The study of the demographical data indicated that the four product owners (A, B, C and D) managed different products. Having the name of the respondents and the products, the identification of the PdMs and the product location was an easy task (see Table 8). Products A and B are developed in a collocated way and the PdMs are collocated with the development team. Project C is currently developed in a collocated way[3] but the PdM is geographically distributed. The development of Project D is distributed between Sweden and Russia.

The focus of this thesis is to determine the agility of the company in distributed environments. For this reason, products A and B were discarded for analysis since they were co-located and not distributed.

PdM	Location	Location of the development of the product
A	Sweden	Sweden
B	Sweden	Sweden
C	Sweden	Denmark
D	Sweden	Sweden and Russia

Table 8. PdMs and products location

2.2.3.2 Quantitative data analysis

Each PdM data was studied individually since they manage different products and as a result they may have different perceptions of the process model, tools and satisfaction.

[3] Product C was initiated as distributed between Russia and Denmark, but then it was brought together in one place, in Denmark.

2.2.3.2.1 Current Documentation Overproduction

Table 9 shows the scores for the two items on Current Documentation Overproduction. Both PdMs consider that the development process generates more documentation than needed, i.e. the documentation is perceived as heavier than it is expected to be. Heavy documentation is not considered agile so further research on documentation overproduction was carried out.

Current Documentation Weight is more than needed	
PdM	***Rate***
C	*Agree*
D	*Agree*

Table 9. Current Documentation weight

2.2.3.2.2 Current PdM Feedback Integration Grade

Table 10 shows the scores for the two items on PdM feedback integration. One of them reported a full ability to integrate his/her feedback in the final product. The other one was undecided. The agility of this variable is improvable in the case of D

Feedback integration was considered in the PMPAT. It could be integrated in three different stages of the development phase (see section 2.1.3.2.5). This feedback is included in the backlog and accepted by all of the parts involved in the development process for release. This collides with the results shown in this variable. The problems of this variable could be caused by the lack of communication between PdMs and developers due to the distribution. Further investigation was done to identify the root causes of the conflicting perception of PdM feedback integration.

Current PdM Feedback Integration Grade	
PdM	***Rate***
C	*Undecided*
D	*Disagree*

Table 10. Values for PdM Feedback Integration Grade

2.2.3.2.3 Communication Speed Satisfaction

Table 11 shows the scores for the two items on Communication Speed Satisfaction. PdM C considers that the communication speed is slower than he/she would like to. D does not agree with this statement. The agility of D is good, but the one from C is very low. This is surprising since product D has geographical, temporal and socio-cultural distance while C only has geographical distance,

Communication Speed Satisfaction	
PdM	***Rate***
C	*Agree*
D	*Strongly Disagree*

Table 11. Values for Communication Speed Satisfaction

2.2.3.2.4 PdMs Process Nature Perception

Table 12 shows the scores for the four items on Incremental and Iterative Development Process. PdM C considers the development process incremental and iterative. PdM D was undecided in the incremental variable but considers the process iterative.

PdM C states that his/her product follows a feature driven development process assigning one feature to more than one team. PdM D does not know if a product is split in different features, but he/she thinks that a feature is assigned to just one team.

The PMPAT states that the agile processes should be iterative and incremental. In terms of development processes product C could be considered more agile than D. PdM D expresses a lack of understanding of the development process. Clarification and a good definition of the development process are advised.

PdM	Incremental Process	Iterative Process	Feature Driven Process	Feature assigned to one team
C	Agree	Agree	Yes	No
D	Undecided	Agree	I don't know	Yes

Table 12. Values for PdMs Process Nature Perception

2.2.3.2.5 Current vs. Desired Progress Measurement Way

Table 13 shows the scores for the two items on the current and the desired way of progress measurement. PdM C cannot measure the process by demonstrations but he/she was undecided about the preferred way of measuring the progress.

PdM D can measure the progress by working software or demonstrations and he/she is satisfied with this measurement progress way.

The PMPAT states that there should be sessions in which the whole team should present the progress by demonstrations. The way of measuring progress of product D follows the specification, but the one on product C seems not be following it.

Current and Desired Progress Measurement Way		
PdM	Rate (C)	Rate (D)
C	Strongly Disagree	Undecided
D	Agree	Strongly Agree

Table 13. Values for Current and Desired Progress Measurement Way

2.2.3.2.6 Changes in Requirements

Table 14 shows the scores for the two items for the questions related to changes in requirements: 1. Developers welcome changes in the requirements, even in the later steps of the development process, 2. Requirements change dramatically between iterations and 3. It is very common to have changes in requirements in later steps of the development process

Changes in requirements			
PdM	Rate (1)	Rate (2)	Rate (3)
C	Agree	Strongly Disagree	Strongly Agree
D	Disagree	Disagree	Undecided

Table 14. Values for Changes in Requirements

PdM C states that developers welcome changes in requirements. He/she considers that there are not dramatic changes between iterations and recognizes usual changes in the later steps of the development process. Further investigation on this last issue should be carried out. PdM C has a good level of agility in this variable, but the usual changes in the latest steps of the development process hinder this agility. This should be improved to achieve a higher level of agility.

PdM D states that developers do not welcome changes in requirements. He/she considers that there are not dramatic changes between iterations but recognizes some changes in the later steps of the development process. The agility level of D is improvable in this variable.

PMPAT states that requirements have to be managed along the whole iteration. There could be dramatic changes between iterations, but it should not be common. Also, there should not be big changes in requirements in later stages of the development process if it is incremental. In conclusion, feedback from the PdM seems not being well processed and managed during the development process.

2.2.3.2.7 Trust

Table 15 shows the scores for the two items for the questions related to Trust. PdM D feels trusted, but PdM C was undecided. Trust is good for both GSD and agile practices. Further research should be done to determine if PdM C feels trusted.

Trust	
PdM	***Rate***
C	*Undecided*
D	*Strongly Agree*

Table 15. Values for Trust

2.2.3.2.8 Capability to Play the PdM Role

Table 16 shows the scores for the two items for the questions related to Capability to Play the PdM role: 1. I am able to attend to every meeting in which software pieces are demonstrated, 2. I am able to coordinate with the development team (without any problem) a prioritized list of requirements every time we have a meeting and 3. I am able to provide support when a development team needs me.

Capability to play the PdM role				
PdM	***Rate (1)***	***Rate (2)***	***Rate (3)***	***AVG***
C	*Strongly Disagree*	*Disagree*	*Strongly Agree*	*2,15443469*
D	*Strongly Agree*	*Strongly Agree*	*Strongly Agree*	*5*

Table 16. Values for Capability to play the PdM role

PdM D is able to attend the meetings for demonstrations, to provide support to the development team when needed and to coordinate with the development team a list of prioritized requirements. He/she is able to play the PdM role in a prescribed way.

PdM C reported some difficulties to play his/her role. He/she is unable to attend to meetings in which software is demonstrated. PdM C is unable to coordinate a prioritized list of requirements with the development team. However, he/she is able to provide support when needed.

PMPAT states that a backlog should be planned and accepted by every part, prioritizing requirements, at the beginning of each iteration. This is not being accomplished in the case of PdM C, who is behaving as the anti-PdM. PdM D is able to play his/her role in a prescribed way. PdM C is less agile than D. PdM D seems to be totally agile.

2.2.3.2.9 Willingness to Prioritize Requirements

Table 17 shows the scores for the two items for the questions related to Willingness to Prioritize Requirements: I would like to take into account the development team to establish a prioritized list of requirements.

PdM D would like, and actually likes (see section 2.2.3.2.6) to take into account the development team to coordinate a list of prioritized requirements. PdM C would also like to change his/her current ability to be able to coordinate prioritized requirements lists with the development team to a more agile way.

Willingness to Prioritize Requirements with the Developers	
PdM	**Rate (1)**
C	*Agree*
D	*Agree*

Table 17. Values for Willingness to Prioritize Requirements

2.2.3.2.10 Current Process Understanding vs. Desired Process Clarity

Table 18 shows the scores for the two items for the questions related to Current and Desired Development Process Understanding: 1. I have clear understanding of the development process structure, 2. The process development is well defined and explained, 3. The process development is well understood and 4. I would like to have a more explanatory and defined development process definition.

Both PdMs agree on having a clear understanding of the development process. PdM C does not consider that the development process is well defined and explained. Both of them agree that the development process is not well understood. Both of them were undecided on having a more explanatory and defined development process definition. Further research on the process model definition and circulation was conducted.

Current Process Understanding and Desired Process Clarity				
PdM	**Rate (1)**	**Rate (2)**	**Rate(3)**	**Rate (4)**
C	*Agree*	*Disagree*	*Strongly Disagree*	*Undecided*
D	*Agree*	*Undecided*	*Agree*	*Undecided*

Table 18. Values for Development Process Understanding and Clarity

2.2.3.2.11 Frequency of F2F Meetings

PdM meetings were also analyzed to check whether or not the Product Manager was considered part of the development team. As the question was open, the answers were classified by the scale shown in Table 19.

1	*Never*
2	*Almost Never*
3	*One or two times per year*
4	*Monthly*
5	*Weekly*

Table 19. Legend for the frequency of meetings with the Product Manager

Table 20 shows the scores for the two on Frequency of F2F meetings with the development team.

Face to Face Meetings Frequency	
PdM	**Rate**
C	*One or two times per year*
D	*Never*

Table 20. Values for F2F Meetings

PdM C meets the development team at least one or two times a year, but PdM D never meets the whole team. This is not surprising since the project is distributed but it is advisable to have one face-to-face meeting at least once or twice a year for a better agility. This collides with the PMPAT since it states that the whole team should have a daily meeting.

2.2.3.3 Qualitative Data Analysis

2.2.3.3.1 Tools

In this section the tools the PdMs consider necessary for a successful development of their products are presented. The data retrieved for this section comes from the question "Please, state five tools you consider essential for the success of a distributed project". In this question it was requested to write the tools in order of importance.

Table 21 shows the gathered data about the tools considered important for the success of the product development. The first column represents the PdM identification and the first row represents the tools reported by the PdMs as essential. Each tool is related to the PdM that reported it by a number that indicates the importance he/she gives it (1 taking the highest importance, 5 taking the lowest). However, PdM C reported four tools and PdM D reported three tools.

PO / Tool	Videoconference	Email	Phone	F2F meetings	Document Handling System	Chat
C	3	2		1	4	
D		1	3			2

Table 21. Tools considered important

PdM C considers F2F meetings most essential. After that, he/she considers Email in the second place, Videoconference in the third and Documentation in the last place. PdM C has a good vision of the essential tools for a successful GSD [42].

PdM D considers in the first place Email, after that Chat and Phone in this order. As this project is distributed, it is remarkable that the PdM does not consider the videoconference tool as essential. F2F meetings were also not included in his/her answer. The vision of this PdM about the essential tools is a bit deviated from suggestions of essential tools reported in other case studies for a successful GSD [42].

2.2.3.3.2 Roles

In the questionnaire there were some questions asking the PdMs to select the role they were playing. They were requested to select which roles they identified in the development process among some different options extracted from literature [53] based on a software feature-driven development process. The question was in form of a checklist with multiple options from the following list taken from [53]:

- External Product Manager
- Lead Product Manager (Collects the requirements of various external Product Managers and prioritizes them)
- Team of product owners
- Feature Manager
- Feature Team
- Coach (ensures that the process the team is following is the best one)

Table 22 shows the results of the roles the PdMs identified in their team and the roles the PdMs think they play. The first row shows the PdMs IDs. The first column shows the most common roles inside Distributed Agile Development [53].

The results indicate that PdM C has a clear understanding of the role he/she is playing inside the development process. PdM D reported to be playing all the roles of the team. In addition, all the roles depicted in the PMPAT were reported to be present in the product managed by PdM D. Further investigation on roles blurriness and clarification was conducted.

PO / Role	PdM C		PdM D	
	Played	Identified	Played	Identified
External Customer			X	X
Lead product Owner	X	X	X	X
Product Owner	X	X	X	X
Team of Product Owners			X	X
Feature Owner			X	X
Feature Team		X	X	X
Coach				X

Table 22. Detected and played roles.

2.2.3.3.3 Communication Protocols

A question to determine the communication protocols between PdMs and developers was also included in the questionnaire: "Please, describe the communication process with the development team (frequency of contacting the team, tools you use for it, face-to-face meetings, protocols, etc)". The results are as follow:

- PdM C: "E-mails and face-to-face meetings in the corridors are the main communication paths."
- PdM D: "I work with collocated and remote developers.
    - **Local developers:** There is communication more or less every day by telephone or F2F meetings.
    - **Remote developers:** There is communication more or less every day by email and MSN. If needed, we also communicate via telephone."

As it can be seen in the answers listed above there seems not to be any communication protocol defined. Communication is started depending on the needs of the project at the specific moment.

Agile relies on face-to-face communication [41]. This is not always possible in a globally distributed environment. A good substitute for face-to-face communication is the use of videoconferences which have been reported as the richest electronic media [42]. However, videoconferences take much time, so abusing the use of this communication way is also not productive. Videoconferences are advised to be used in team meetings, e.g. Scrum or kickoff meetings [54]. In the case of PdM D, since the project is distributed, the use of video conference tools is recommended [55].

2.2.3.3.4 Changes in requirements

Quantitative analysis has shown some aspects of the management of changes in requirements and the attitude of the development team towards those changes. This section sheds more light into the reasons why these changes occur. The respondents were requested to "describe the main reasons of changes in requirements during the development process (Why they appear? When?, etc)". The answers to this question are as follow:

- **PdM C:** "New functions or features are sold by sales at any time. There are also changes due to bad understanding of Product Manager needs."

- **PdM D:**
    - **Changes in existing requirements:** "Developers discover things that are not possible to be done. This happens during the demonstration (not detected reading a document). Sometimes I discover something that is not very usable".
    - **Adding or removing complete requirements:** "A new requirement can be added if an error is discovered or if there is some time left before testing. Requirements can be removed if there is not enough time left before testing starts."

In conclusion, changes in requirements come because of the inclusion of new features, inconsistencies or misunderstandings with between the PdM and the development team. Most of the causes are the usual ones, but the bad understanding of requirements. These misunderstandings could be caused by the lack of contact between the PdM and the development team.

2.2.3.3.5 PdMs Reported Improvements

In the questionnaire there was an open question designed to capture the PdMs recommended improvements over the current development process. The question was "If you could change something from the development process you are currently following, what would you change?" The next list contains the replies to this question by PdM:

- **PdM C:** "Clearer alignment of roles and deliverables with the defined process development specification utilized in Gothenburg with the rest of the global organization."
- **PdM D:** "I would prefer to have more developers located at the same office as I am"

In conclusion, PdM C does not consider the alignment of roles to be clear enough. PdM D is not satisfied with the developers' distribution. It seems that PdM D would like to be able to be more involved in the development process by having a less distributed team.

2.2.4 Summary

This section summarized the findings of the survey conducted in the company involving product managers from Sweden. The resulting data analysis detected very interesting findings that leave some place for improvements:

- The weight of the documentation was not perceived as light.
- One of the PdMs considers that the feedback he/she provides is not fully integrated in the final product. This collides with the PMPAT, so it was also addressed in the next research steps.
- In one product, the communication with the development team is reported not to be agile enough.
- One of the PdMs reported that his/her process follows an iterative and incremental process model. The other PdM have reported the opposite, expressing also a lack of understanding of the development process nature.
- One PdM reported being able to measure the product progress by working software or demonstrations while the other one reported not being able to do it. This is tied to feedback management inside the development process, so it was further studied.
- One of the PdMs reported to have frequent changes in later stages of the development process while the other one reported the opposite. Late changes are not supposed to

happen if following the PMPAT since the changes should be reported and integrated in an incremental way.

- One of the PdMs reported to feel fully trusted by the development team, while the other one was undecided.
- One respondent reported not being able to play his/her role in an agile way.
- F2F meetings were identified as essential among most of the respondents. Phone and email are sometimes considered more important than videoconferences or F2F meetings.
- The frequency of meetings with the whole development team ranges from Never to 1 or 2 times a year. This could be affecting the agility of the rest of the variables. It also collides with the PMPAT since team meetings should be held daily.
- One PdM seems to not fully understand the role he/she is playing.
- Communication protocols seem not to be defined. Communication is established when it is considered to be needed and/or it is requested, what actually is the agile way.

In conclusion, both PdMs demonstrate some agility features, and they are willing to be more agile. In the next research steps the author of this thesis focuses his investigation on causes of documentation overproduction, a not full PdMs' feedback integration and causes of blurriness on roles and on the nature of the development processes.

2.3 Final Conclusions on both Surveys

Employees with different roles may have different perspectives of the same event inside an organization. For this reason the author of this thesis aimed to have the opinion from two different roles for a similar group of variables.

Different challenges on the way to agile were detected from the analysis of the two surveys:

- In both surveys, the respondents reported that the documentation is not lightweight. This is not considered agile, so further investigation was done to detect what is the root cause of this documentation weight.
- The PMPAT considers agile processes as both incremental and iterative. However, some respondents from both PdMs and developers have reported not following iterative and/or incremental development processes.
- Most developers reported to usually integrate the PdMs' feedback in the development process, but they also reported that changes in requirements are not allowed in every step of the development process. One PdM reported that the feedback is not fully integrated in the final product. Further investigation was carried out to find the root cause of this issue.
- Most developers would like to discuss requirements alignment with the customer. However, one PdM reported not being able to attend to demonstrations and team meetings. The PMPAT states that the PdM should be involved in the development process and attend to daily meetings, which is also not achieved. This could also solve the detected lack of understanding of the development process detected in one PdM.
- In some cases the development process is blurred. Further investigation on development process documentation, clarification and circulation was done.
- Roles blurriness has also been detected in developers and PMs. Further investigation on clarification of roles and responsibilities was done.

- Good intra-team trust and motivation was detected from the developers' side, but only one PdM reported to feel trusted by the development team.
- PdMs consider communication tools essential for a successful development of their product but developers are more focused on tools regarding coding, development and sharing. Email and telephone are prioritized over video-conferences.

In conclusion, both roles have some agility features and they are willing to be more agile. Thus, the company is prepared to be agile. However, there is room for improvement. The next research step is focused on analyzing the root causes of the detected challenges: documentation overproduction, process and roles blurriness, and incomplete feedback PdMs integration.

3 HINDRANCES: ROOT-CAUSE ANALYSIS

3.1 Interviews Design

This step, as the previous ones was designed in an incremental way. The first interview was designed according to the findings of the surveys and the second interview according to the findings of the first interview (see Figure 17, left side).

The interviewees were informed about the nature of the study and the previous findings before starting the interview. Thus, different levels of causes could be established (see Figure 17, right side). This allowed the author to get a deeper perspective on the root causes of the detected challenges, being able to provide more suitable information.

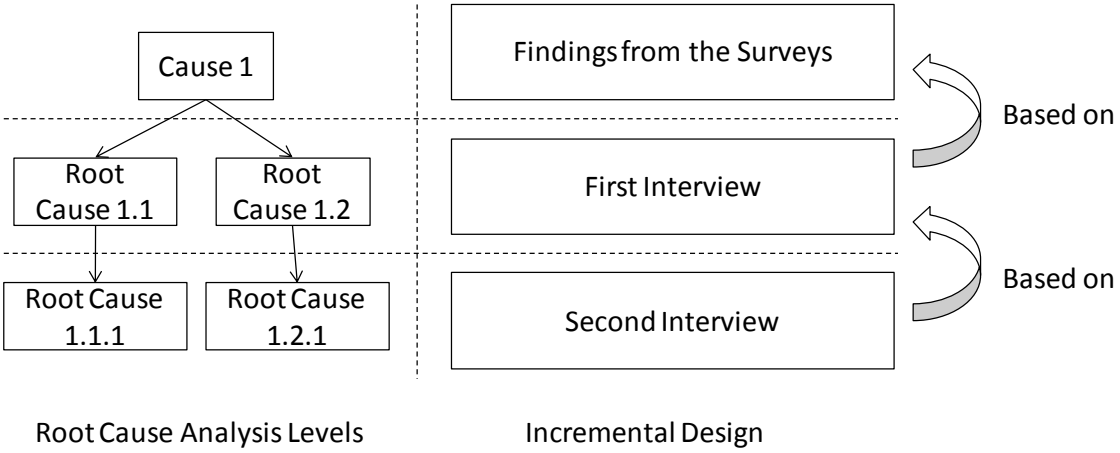

Figure 17 Interviews Design

3.2 Interview Questions Formulation

The interview questions were formulated with the purpose of finding the root causes of the main issues detected in the surveys:

- Cause 1: Why feedback from the Product Managers is not fully integrated?
  - **Developers**: "*More than 50% of the respondents reported to always integrate Product Managers' feedback. The Product Managers' Feedback Integration Allowance variable does not have enough level of agility, but the Feedback Integration Completeness looks quite agile, with some exceptions*."
  - **Product Managers:** "*One of the PdMs considers that the feedback he/she provides is not fully integrated in the final product. This collides with the agile practices*."
- Cause 2: How are the development processes defined? Is there any specification?
  - **Developers**: "*The number of incremental and iterative development process responses is much reduced. Thus, the agility is low since the PMPAT consider a process agile when it is both incremental and iterative. The undefined responses indicate that the development process nature is blurry for some developers.*"
  - **Product Managers:** "*One of the PdMs reported that his/her process is incremental, iterative and feature driven, what means a high level of agility in*

the process nature. The other PdM have reported the opposite, expressing also a lack of understanding of the development process nature."

- Cause 3: Why is there documentation overproduction?
    - **Developers**: "*The agility on documentation is half-way. The documentation should be lighter. However, the developers do not express desire of change in this aspect.*"
    - **Managers**: "*Both PdMs reported that the documentation is heavier than expected. This was considered in next research steps since heavy documentation is not considered agile*."
- Cause 4: Why roles are blurred and not well understood?
    - **Developers**: *"Roles involved in the development process seem not to be well understood by the respondents"*
    - **Managers**: *"One PdM seems to not fully understand the role he/she is playing."*

The interviews were semi-structured combining open-ended, structured and exploratory questions. The structured questions required a specific answer while the open-ended questions provided the interviewees to express their own opinion. Both interviews were conducted face-to-face.

The results were extracted through a thematic analysis [56,57]. Data was analyzed by theme, which, in the particular case of this research, means by risk for agility. In this way, the root causes of each hindrance for agility were determined.

3.3 Unit manager Interview (from Sweden)

3.3.1 Design

The two hours interview was planned in the following way:

1. Providing an explanation of findings and purpose of the interview. (30 minutes)
2. Getting some knowledge on the development process nature and the available documentation. (30 minutes)
3. Getting some knowledge on the requirements management and feedback integration tasks inside the development process. (30 minutes)
4. Getting and discussing any other extra information resulting from the elicitation process. (30 minutes)

Table 23 shows the questions formulated for this interview.

Interview Questions	
Generic Exploratory Questions	
1.	What can you tell me about the nature of the development process (capture the picture of the development process)?
2.	What are the Product Manager needs in terms of warranty costs? What are the Product Manager needs in terms of number of projects per year?
3.	Is there any development process specification defined for the process you are following?
Exploratory Questions on Development Process Nature	
4.	Could you please enounce and describe the phases of the development process of the selected system (from beginning to end)?
5.	Could you please identify what pieces of documentation are composed and in which phase?
6.	Some Product Managers and developers have reported overproduction in documentation. Do you agree with this? Why do you think this happens? Can you please suggest where extra documentation is being generated inside the development process?
7.	What are the inventory or queue points in the development process?
Exploratory Questions on Requirements and Feedback management	
8.	What is the procedure to elicit requirements from the Product Manager (internal customer)?
9.	When do you realize that the team is not going to meet the requirements on time? How do you realize?
10.	What is the frequency the Product Manager gives some feedback? How is that feedback collected and documented? Who is the responsible?
11.	Some developers and Product Managers have reported that the Product Managers feedback is not fully integrated in the development process, what do you think about this statement? Why do you think this occurs?
12.	Is the integration of the feedback from the Product Manager taken into account in that development process specification? In which part of the development process?
13.	Changes in latest stages of the development process have been reported by Product Managers and Developers: Is feedback gradually integrated?
14.	Are there a lot of misunderstandings between Product Managers and Developers in requirements? How are they identified?
15.	How is the ceremony in meetings with the Product Manager (software demonstrations and list of prioritized requirements)?

Table 23. Questions of the first interview.

3.3.2 Results

The results from the interview are summarized as following:

- The interviewee agreed on the heavy weight of documentation but he considered it necessary for the maintenance of products. As a result, the documentation weight varies depending on the product.
- The interviewee reported that there is a development process specification available online for all the company's staff. The CMMI Level 3 certification obliged the company to generate a lot of documentation for the development process specification. Personnel from different locations were involved in the documentation process so the specification is perceived to be well known and circulated.
- The interviewee agreed that having changes in later stages of the development process is a critical issue. He suggested the implementation of these changes in earlier stages of

the development process by requesting the PdMs feedback sooner. Since the PdMs are distributed from the development team, this is not an easy task.

- The interviewee explained that developers and PdMs have gone through different development process models since these process model have been changing over the time. He suggested that this could be the main reason why there is blurriness in the development process.
- The interviewee agreed that the progress of some products are not being measured by demonstrations or working software. This seems difficult since the distribution does not allow all the development team to be present in the demonstrations. He considered this as a future potential improvement for achieving a higher level of agility.
- The interviewee reported that developers do not prioritize a list of requirements with the product manager; the project manager does it instead. This could be caused because of the distribution of the PdMs and the easiness of sending just one person instead of the whole development team.
- The interviewee explained that the PdM role is not well defined and understood since there have been lots of changes in roles in the last years.
- The interviewee identified the testing phase as a bottleneck. The testing is not done in small pieces; it is pushed until the end of the development process instead. In his opinion this is caused by the dislike of people for testing.

In conclusion, the heavy weight of documentation was needed to achieve the CMMI certification and to maintain old products. Testing should achieve a higher level of agility to avoid bottlenecks and changes in later stages of the development process. Building small pieces of working software could help to reduce the testing bottlenecks and to demonstrate the progress to the PdM.

There is a development process specification available for all the personnel of the company. However, there is blurriness in the perception of the development process nature. The lack of feedback integration is still unclear since the respondent did not give any conclusive answer. Roles shall be well defined and understood among the personnel. Further investigation to understand the root cause of these facts was carried out in the next research steps.

3.4 Developer Interview (from Russia)

3.4.1 Design

The interviewee was informed about the conclusions extracted from the previous interview. The interview was focused on the not full PdMs feedback integration and the blurriness of the process models. The other causes were considered out of scope or irrelevant after the interview with the unit manager. These causes were explained to the interviewee as well as the focus of the study.

The one hour interview was planned in the following way:

- Explaining of the findings and purpose of the interview (10 minutes)
- Getting some understanding of the perception of the interviewee about the development process (10 minutes)
- Trying to understand why the process model is blurry (20 minutes).
- Getting some information of the perception of the respondent with respect to the feedback integration (10 minutes)

- Getting any other extra information resulting from the elicitation process (10 minutes).

Table 24 shows the questions for the second interview.

Interview Questions	
Generic Exploratory Questions	
1.	What is the name of your product?
2.	What development process you follow?
Exploratory Questions on the Process Model Understanding	
3.	I would like to have more information about the development process.
4.	Is there any development process specification that you follow?
5.	Do you know that there is a development process specification?
6.	Do you know where to find the development process specification?
7.	What do you think about the development process specification? Is it heavy?
8.	Do you find useful to have a development process specification?
9.	Do you get lost in the development process?
10.	What are the main steps you identify in the development process?
Exploratory questions on the PdM Feedback Integration	
11.	How often do you receive feedback from the PdM?
12.	How often do you have meetings with the PdM?
13.	Do you find feedback from the PdM useful?
14.	Do you know any specification on how and when feedback from the PdM should be integrated?
15.	How often do you integrate feedback from the PdM?
16.	How often do you communicate with the PdM?
17.	How do you communicate with the PdM? And vice versa?
18.	What is the attitude of the team towards the PdM feedback?
19.	Do you think that the feedback you receive from the PdM is enough for the success of the product development?
20.	Do you think that the PdM feedback is clear enough?
21.	Which communication way do you consider essential for the success of the product development among the following: F2F meetings, email, phone or vide-conferences?

Table 24. Questions of the Second Interview

3.4.2 Results

The results of the interview are summarized as following:

- The interviewee reported to not be taking into account the development process when working. He does the tasks he is assigned to do.
- The interviewee knew that he is following some development process but he did not know which.
- The interviewee knew about the existence of a software process specification. He overviewed it in the very beginning but he does not use it. He considered it heavy but not difficult to read.
- The interviewee considered good to have a development process specification since "*it is good to know how the process is structured*". However, the development process is totally transparent for him, i.e. the developer is abstracted from the development model and performs the tasks he is assigned without needing to know in which phase of the development process he is.

- The interviewee did not know in which phase of the development process he was working.
- The interviewee confused the terms “Product Owner” and “Product Manager” and did not recognize the last one, which is the one defined in the PMPAT.
- The interviewee reported that the developers are not always present in demonstrations. It depends on the product and the product manager. This could be caused by the geographical distance between developers and PdMs. The developer did not express any interest in being in a meeting with a PdM. This collides with the PMPAT since the whole team should be present when the demonstrations are done, demonstrating the part they have developed and explaining its functionality.
- The interviewee reported that the feedback from the product manager is not constant during the development process, i.e. in the beginning there is feedback once per week and later there is feedback once per month. This also collides with the PMPAT since it is designed to retrieve feedback from the PdM along the whole development process.
- The interviewee informed that feedback is integrated depending on the deadlines. The integration is agreed and prioritized with the product manager. The not integrated feedback is normally included in the next release of the product. This matches with the PMPAT, but still rise a question on why one PdM reported his/her feedback not to be fully integrated. It was considered as dissatisfaction from the PdM side.
- The interviewee knew that there is a protocol to integrate PdM’s feedback. He pointed out that sometimes this protocol is changed depending on the needs of the product and the team in a specific moment. However, this should be informed to Division and previously approved by every part, which is not done.
- The interviewee reported that delays in meeting requirements are normally produced by obstacles caused by the distribution, e.g. new equipment to be sent from one location to another.
- The interviewee considers videoconferences and mails essential for the success of the development of the product due to the distributed environment.
- The interviewee accepts the feedback from the PdM with resignation.

In conclusion, the possible root cause of the lack of knowledge about the development process could be the abstraction of developers from the development process specification. As the tasks are imposed and not discussed, the developers do not need to know in which phase of the development process they are. This goes against the PMPAT which clearly states that the developers should be present in every phase and should take part in the decisions.

The PdM role seemed to be not well understood among the respondents in both surveys and it was not well understood by this interviewee. A definition and documentation of roles shall be designed and circulated.

The possible root cause of the reported lack of PdMs feedback integration could be the dissatisfaction of PdMs with the partial postponing of this feedback. Further investigation should be carried out regarding this last challenge.

4 TOWARDS BECOMING AGILE: PROPOSED SOLUTIONS

4.1 Identifying sources of information

In the light of the findings of the previous research steps, the author decided to provide some recommendation on models, tools and best practices.

Following the first step from [46], the author performed a search to look for related systematic literature reviews. This was done to determine the need for a systematic review. The search string that was used for this search was: *("systematic review" or "literature review") and ("global software engineering" or global software development" or "distributed software engineering" or "distributed software development") and ("best practices" or recommendations or tools).*

The search was executed over Scopus, Springer and Compendex databases. The results provided three potential studies covering the needs of this thesis [6,24,39].

The quality, the coverage level and the fulfillment of the studies were ensured through a checklist extracted from [58]. The questions included in the checklist were the following:

- What are the review's objectives?
- What sources were searched to identify primary studies? Were there any restrictions?
- What were the inclusion/exclusion criteria and how were they applied?
- What criteria were used to assess the quality of primary studies and how were they applied?
- How were the data extracted from the primary studies?
- How were the data synthesized? How were differences between studies investigated? How were the data combined? Was it reasonable to combine the studies? Do the conclusions flow from the evidence?

The findings from the execution of the checklist over the three systematic reviews showed that the most suitable one for the study was [24]. It studies process models, challenges, best practices and tools in GSD, covering the recommendations needs.

There were several limitations that were considered when selecting this study as main source of the assessment:

- The age of publication of this paper was 2009 (see top light grey part from Table 25.
- It lacks Springer as search database.

To evaluate the impact of these lacks the author performed a search of the potential studies left (see Table 25). The process followed to perform this extension was to reproduce the search included in [24], constraining the ages of publication from 2010 to 2011 (see top part dark grey part from Table 25). The search was also expanded to Springerlink database since it was available from BTH (see light grey part in Table 25).

← 2009	2010-2011
Springer	**Covered Databases+Springer**
Search on DB	Search on publication year

Table 25. Reproduction of the search of the study

The results of the two reproduction parts can be seen in Table 26 and Table 27. As it can be observed, the resulting set in both cases is small to discard the study as source for assessment and to consider the constraints as a research threat. Even if an extension of [24] was not considered necessary, it is advised. In this thesis this is not included due to time restrictions.

Table 26. Extension of [24]

Database	IEEExplore	ACM	Science Direct	Compendex	2010-2011
Retrieved Results	10	6	100	55	
Potentially Relevant Results	1	0	3	6	

Table 27. Extension of [24] (2)

Database	SpringerLink
Retrieved Results	14
Potentially Relevant Results	3

Despite the several different hindrances for agility that have been detected in this case study, the recommendations are focused on the ones that are considered to be causing a higher impact in the agility of the development process. These are: blurriness of PdM role, visibility of the development process and PdM feedback integration. Recommendations over other issues are also provided, but these last are the main focus.

4.2 Recommendations on Roles Definition

One of the important aspects for the success of a distributed team is the implementation of an effective team and organizational structure [7]. To establish an organizational structure is the creation of roles, relationships and rules, which can facilitate effective communication, coordination and control. Authors from [59] recommend the documentation and circulation of this structure. The purpose of this exercise is to make all the staff understand roles and responsibilities within the project [7].

Authors from [60] explain that the splitting of their system into subsystems (products) was accompanied by a partitioning of the roles and responsibilities caused by the disjoint locations. They report to have overcome this challenge by defining the new definition of roles and responsibilities and publishing it in the corresponding web page of the associated products.

The author of this thesis recommends to clearly defining the different roles that may take part in the agile process model and their corresponding responsibilities. The definition of roles is included in the PMPAT, but the responsibilities were not detected to be clearly defined. This definition should be completed by assigning these roles to concrete persons per product. This could be accessible and visible through a product web page to circulate this information through the different locations taking part in the product definition. The availability and the content of this information should be communicated in a project meeting.

4.3 Recommendations on Feedback Integration and Project Progress Measurement

Authors from [61] created a central team. This central team was responsible for the requirements, software architecture and some aspects of design, system test, integration, project management and defining the overarching processes. The other teams were assigned an individual to manage the interaction between the central team and the rest of the teams. This responsible and the team had a weekly meeting conducted in a structured format. The questions formulated by the team were required to be submitted before the meeting was held. In this way, the central team answered in a consistent manner and not on the fly. The different features were developed simultaneously in iterations integrating the feedback from the supplier manager. When the iteration was finishing, the teams sent the product to the central team to be tested and integrated.

Authors from [60] followed a development process based on load builds. These load builds were sent to testing and integration periodically so that the feedback from the PdM was always integrated and changes were implemented in a consistent way. Even if literature suggests daily load builds [41], they recommend doing it weekly.

Previous results of the case study revealed that PdM's feedback is usually integrated in the final product, but sometimes it is delayed for next releases agreeing with the customer on this decision. The study has also reported frequent and big changes in late stages of the development process. This could be solved by asking the PdM for feedback in earlier stages of the development process for integrating it as soon as possible, as the first interviewee suggested in section 3.3.

The recommendations from the author with respect to the PdM's feedback integration are:

- To run weekly scrum meetings with the PdM instead of daily. This could help to reduce the geographical distance between PdMs and developers. If it is held weekly developers and PdMs could feel the necessity for this meeting and the attendance rate could increase.
- To design a protocol or format for the meeting and include it in the PMPAT (it is currently not included). This has a potential to lead the team straightforward to the main issues, helping them to detect risks, bottlenecks and inconsistencies and aiding them to get the feedback from the PdM.
- To have a shared document with the PdM in which the development team could formulate their questions. This could help the PdM to prepare his/her feedback in advanced in a consistent manner. This could also increase the agility of the PdM and the feedback consistency. It could also help to reduce the temporal and geographical distance since it is considered asynchronous computer mediated interaction. Thus, the feedback integration might be more agile.

The authors from [62] also suggest publishing the project progress in a web-based portal, so that any member of the project can consult the overall progress. This could make the PdM more comfortable so that he/she may tend to be more proactive increasing the agility of the process by providing more feedback in earlier stages. It could also serve to show the velocity of the development process and to indicate the PdMs where changes can or cannot be implemented.

4.4 Recommendations on Process Visibility

A certain lack of knowledge about the development process was detected. The team members are assumed to be knowledgable about agile practices according to the PMPAT.

Authors from [61] used a tool in one of their projects for change management called Change Control Board. This tool made the developers understand the rationale of the changes produced in the product.

The author of this thesis proposes an electronic board to show in which stage of the development process each product is and which are the next and previous steps. In this way, the staff can always consult in which stage they are. When talking about an electronic board it could also be some sort of application in which the teams could record their progress. In case of collocated development teams, a paper-based burn down chart daily updated at first time in the morning is also an alternative. This could help the manager to check the project progress.

The author of this thesis would also like to suggest explaining the stages of the development process to developers, informing them in which stage they are and which the next step is to be taken.

Authors from [60] gave training to employees on remote locations on products and processes. The PMPAT was composed by people from every location involved in the distribution. However, this does not exactly have to mean that all the developers from every location are aware of and understand the development process. Employees' training could be led by the people that were involved in the composition of the PMPAT from each location. It could increase the agility of the development process execution.

4.5 Summary on Challenges and Recommendations

The analysis of the variables in sections 2.1 and 2.2 reported that the company is prepared to adopt agile. However four main hindrances for agility were detected, i.e. heavy weight documentation, lack of process visibility, lack of contact with the PdM and roles blurriness (see Figure 18). This information came from **surveys** performed to two roles, **developers** and **product** managers.

To detect the root causes of these hindrances an **interview** based root cause analysis was performed (see section 3). Two roles participated in this analysis, a **developer** and a **unit manager**. The interviews revealed the root causes of the hindrances:

- The documentation weight is constrained by the CMMI level 3 certification and the maintenance of old products (see Figure 18).
- The process visibility is may be caused by the abstraction of the developers are from the development process structure (tasks are imposed instead of discussed) and the big amount of changes that the development process has had over time (see Figure 18).
- The lack of PdMs feedback integration may be caused by the lack of contact of the developers with the PdM and the late PdM's feedback request (see Figure 18).
- The roles blurriness may be caused by the poor roles definition included in the PMPAT and the changes roles definition have had over time (see Figure 18).

Some recommendations were provided in section 4 regarding the detected root causes. These recommendations were based on **literature**:

- To avoid the abstraction of developers from the development process three solutions were proposed:
    - Provide an electronic board in which the developers can record a burn chart and see the progress and the stages of the development process.
    - Explain the development process to the employees, previous, current and next stage.

    - Train the employees in the development process specification.

- To mitigate the lack of contact with the PdM three solutions were provided:
    - Perform weekly scrum meetings under a protocol.
    - Share a document with the PdM in which the developers could formulate their questions to the PdM before the meeting.
    - Report the progress of the project in a web based portal to indicate in which part the PdM can report feedback and when is it integrated.
- To avoid the roles blurriness two solutions were proposed:
    - Clear definition of roles, responsibilities and persons assigned to those roles in each product.
    - Make this definition available in the web page of each product.

As it can be seen, the information was retrieved from three different roles (developers, PdM and Unit Manager), providing the study a triangulation of roles. The information was also retried following three different research methodologies, i.e. surveys, interviews and literature (marked with S, I and L respectively in Figure 18) providing the study a triangulation in research methodologies.

Four hindrances for agility were detected, several possible root causes for these hindrances were retrieved and several solutions to overcome the root causes were proposed. The thesis objectives were to determine if the company was prepared to become agile and to propose solutions based on GSD best practices reported in literature. Hence, the thesis objectives are covered.

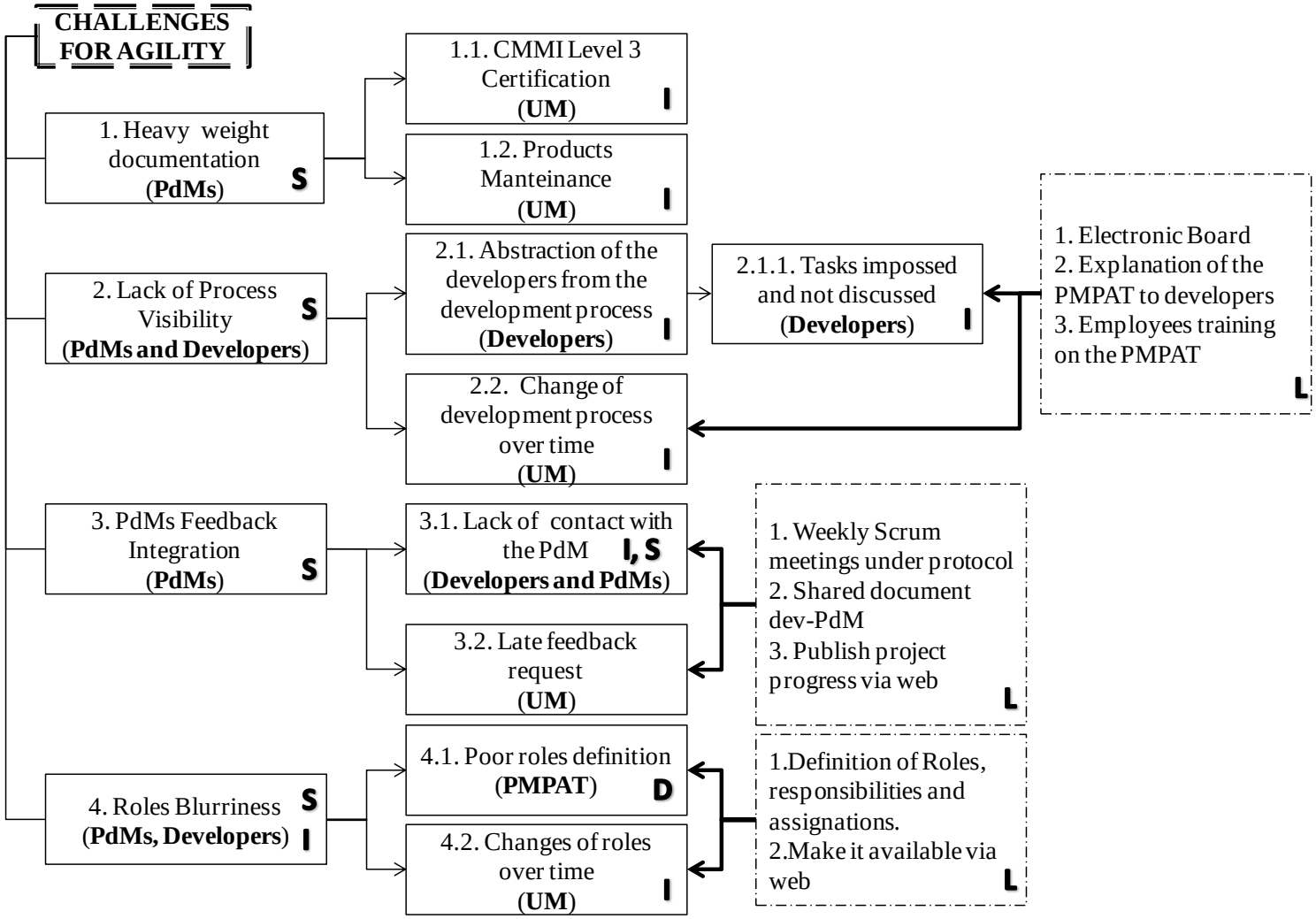

Figure 18. Challenges and recommendations

5 EVALUATION: NOVELTY AND USEFULNESS SURVEY

5.1 Goals of the Survey

This subsection reflects the findings of the last survey conducted in the company. The purpose of this survey is to explore the novelty of the detected hindrances and the usefulness of the proposed solutions. As a result, this survey served to validate the present study.

5.2 Survey Design

The survey was divided in three sections corresponding to three of the four detected hindrances for agility (documentation overproduction was discarded since it was justified as not relevant). Each section was formed by the presentation of each corresponding hindrance and the presentation of the proposed solutions to mitigate that hindrance.

All the questions were formulated in form of multiple-choice. To evaluate the novelty of the detected hindrances the respondent had to choose among three options (see Figure 19 a)). To evaluate the usefulness of the proposed solutions the respondent had to choose among five options (see Figure 19 b)). Every section ended with an open question enounced as following: "*In case you have selected "We will apply this with modifications", or "We will not apply this" in any of the proposed solutions, could you please explain the reasons of your answer?"* With this last question, the respondent could express his reasons and opinions about the proposed solutions.

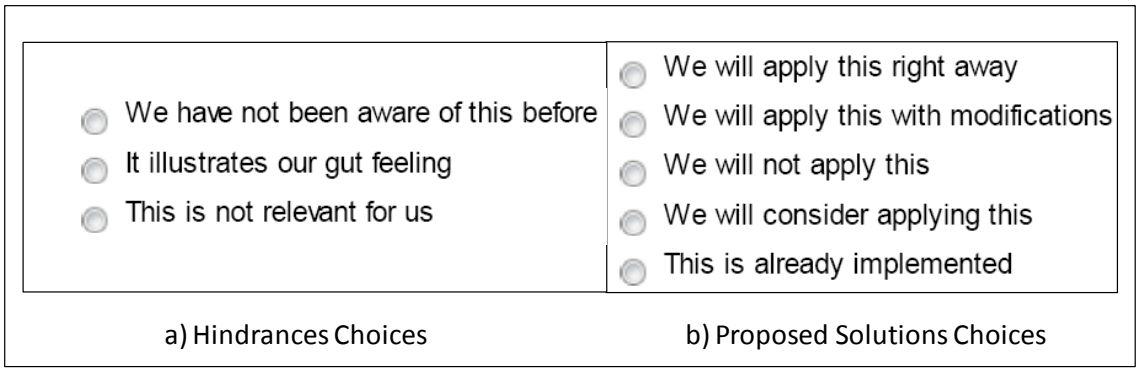

Figure 19. Survey Choices

5.2.1 Survey questions

Based on the results obtained in the study, the survey questions were formulated (see Table 28).

Survey Questions
Hindrance 1
Lack of product manager's feedback integration
Lack of contact between product managers and developers
Proposed Solutions for Hindrance 1
To run weekly scrum meetings with the product manager instead of daily.
To design a protocol or format for the Scrum meeting and include it in the Process Management for Agile Teams specification (it is currently not included).
To have a shared document with the product manager in which the development team could formulate their questions.
Hindrance 2
Lack of process visibility
Proposed Solutions for Hindrance 2
To have an electronic board (physical or web-based) to show in which stage of the development process each product is and which are the next and previous steps
To explain the stages of the development process to developers, informing them in which stage they are and which the next step is to be taken.
To provide the employees training on the Project Management for Agile Teams specification
Hindrance 3
Lack of roles understanding
Proposed Solutions for Hindrance 3
To clearly defining the different roles that may take part in the agile process model and their corresponding responsibilities
To complete the definition by assigning these roles to concrete persons per product.
To make this information accessible and visible through a product web page.
To communicate the availability and the content of this information in a project meeting.

Table 28. Survey Questions

5.2.2 Targeted Population

The survey was designed to be answered by the representative of the company. The respondent was offered three hindrances for agility, which were detected in the thesis together with proposed solutions to mitigate these hindrances.

The web-based survey link was sent to the representative of the company in May 2011 with the following instructions:

"This survey is used to validate the results of the master thesis prepared by David Musat. The purpose of this survey is to explore the novelty of the detected hindrances and the usefulness of the proposed solutions.

Respondents are offered three hindrances for agility, which were detected in the thesis together with proposed solutions to mitigate these hindrances."

5.3 Results

The results of the survey were obtained within the same day it was sent. The response provided very interesting results:

- **Lack of contact with the PdM and lack of PdM's feedback integration:** Both of them were reported to express the gut feeling of the company. The proposed solutions regarding this hindrance had the following response:
  - *Weekly Scrum Meetings***:** They will apply this with modifications. The respondent informed that they already apply daily Scrum meetings in some projects and weekly in some others.
  - *Protocol or format for the Scrum meeting*: They will apply this right away.
  - *Shared Document Dev-PdM*: They will consider applying this.
- **Lack of process visibility**: It was reported to express their gut feelings. The proposed solutions had the following responses:
  - *Electronic Board*: It is already implemented.
  - *Explaining the development process stages*: It is already implemented.
  - *Training to employees*: They will consider applying this.
- **Lack of roles understanding**: It was reported to express their gut feeling
  - *Clearly Defining Roles*: It is already implemented.
  - *Complete the definition by assigning roles to individuals*: They will consider applying this.
  - *Make the information visible per each product*: They will consider applying this.
  - *Communicate the availability of this information*: They will consider applying this.

In conclusion, the three detected hindrances were confirmed and accepted by the respondent, who was aware of them before the study. One of the proposed recommendations will be applied right away and another one will be applied with modifications, both of them for mitigating the lack of contact with the PdM and lack of PdM's feedback integration. Three recommendations are already being applied. Four recommendations will be considered to be applied.

The results conclude that the author did not discover anything the representative was not aware of. Hence, the author provided empirical data that demonstrated what the representative of the company suspected.

The author also proposed several solutions that were considered useful in helping to mitigate the detected hindrances. They are proposed to improve the performance in the future since most of them will be considered to be applied. Hence, the results are not novel but the proposed solutions are useful.

6 DISCUSSION

6.1 Evaluation of the Empirical Data

This case study has reported four hindrances for agility in a global distributed company. With the help of GSD related literature, several solutions to overcome the detected hindrances have been proposed.

It has been noticed that questions asked about the same event differ on the answer depending on the perspective of the respondents, i.e. roles and interests. This match with the findings of the study reported in [63]. Even though, there is a major pattern observed: every individual taking part in the study is willing to be more agile.

The agility willingness was not only reported by the individual about him/herself but also about other roles. In the case of developers, they reported willing the PdMs to be more agile, and vice versa. Also the Unit Manager reported a willingness for all the roles involved in the study to be more agile.

It also has to be taken into account that the willingness to be more agile cannot be generalized to all the employees from the company. The surveys did not have a response rate of a 100%, and even if they had, only a few products were involved in the study. We can say that there is a trend on willing to be agile. The individuals taking part on this thesis are prepared to adopt agile when overcoming the detected hindrances.

The three hindrances affect to communication and coordination of the development process. Each one can be enclosed in each category as follows:

- Lack of communication with the PdM: Affects coordination of changes produced by the PdM feedback.
- Lack of process understanding: Affects both coordination and communication. If a developer do not know the communication process he/she has to follow depending on each stage of the development process, it is very difficult to coordinate with him/her. The communication process will also be slower.
- Lack of roles understanding: Affects to coordination since the developers do not know to whom he/she has to communicate to solve different issues.

The three hindrances can also be classified in the geographical distance challenge category defined in [3] and explained in section 1.1.1.3. Since there is no time difference between Sweden and Denmark, no temporal distance challenge is faced. The time different between Sweden and Russia is two hours. Since it is also not a big difference, the detected hindrances cannot be related to this challenge category. However, Russian, Swedish and Danish cultures are taking part in the same projects. Could agile also help to overcome the socio-cultural distance?

6.2 Human Factors in GSD, can Agile Help?

Cultural factors can negatively impact the performance in GSD environments [64]. In the case study of this thesis three countries have taken part, i.e. Sweden, Denmark and Russia. Having countries with such a big cultural difference taking part in the development process can derive to cultural factors affecting the team's performance. The authors from [65] collected the human factors that most commonly are affecting global organizational environments.

Individualism is one factor that needs to be highlighted when talking about teams, roles and principles. It is defined as the behavior towards the group. A high index of individualism

brings an enjoyment of challenges and expectations of rewards for a hard work, a high valuation of people's time and a respect for the privacy. A low level of individualism brings emphasis on building skills, work for intrinsic rewards and harmony over honesty [65].

Power distance is defined as the way a society handles inequalities. If the power distance index is high, then there are strong hierarchies defined. If it is low, employees and managers are considered at the same level.

Talking about hierarchies it is inevitable to think about roles. Three roles have been targeted in interviews and surveys as part in this study, i.e. product managers, project manager and developers. Table 29 shows the mapping of the studied subjects' roles into their locations.

Country	**Role(s)**
Sweden	Developers, Product Managers and Project Managers
Denmark	Developers
Russia	Developers and Project Managers

Table 29. Mapping of Roles into Locations

The power distance index from Sweden and Denmark is not very high (30 over 100), but the one from Russia reaches a 92 over 100. The Individualism index is also very different in the three countries. Sweden and Denmark have a low index of individualism (14 and 10 respectively) but Russia has a higher one (39-40) [58].

Some products have been detected to have all the developers in Russia and the product and project managers in Sweden and/or Denmark. Subjects with such different power distance indexes can cope [60].

Taking a look at the individualism indexes, it seems evident that the targets of a Russian developer are not likely to match with the targets and the points of view of a Danish developer. Again, the risk is incremented since the pursued targets could be different and the distribution and the geographical distance do not help to overcome this potential challenges.

Some of proposed solutions for the hindrances detected in this study are inspired on agile techniques. But what role do the cultural aspects play in the agile adoption techniques?

- The introduction of agile methods can upset hierarchical organizational structures within a company [66]. This may cause resistance in Russia, which has a high level of power distance.
- Agile methods emphasize the group over the individual [66]. The authors from [66] state that this can constrain the tendencies of countries with individualism trends (Russia), but keeps with high values of collectivism (Sweden and Denmark).

Several research questions can be further investigated as continuation of this study:

- What are the cultural factors that can hinder the agility of the teams combining subjects coming from these three countries?
- Can different indexes individualism and power distance in individuals from the same team affect the team's performance?
- What are the possible solutions to overcome the potential detected hindrances? Can agile help?
- What are the effects of the proposed solutions in each culture?

6.3 Validity Threats

Several validity threats were identified in this thesis. The threats "instrumentation", "processing error" and "generalization of the study", are related to the surveys conducted in the study. The threat "interviewee misgiving" is related to the interview conducted to the developer. The threats "selected literature for recommendations" and "agility indicators" are related to section 4.1.

1. **Instrumentation.**

   There is a threat in the instrumentation of the surveys conducted to developers and PdMs. As it was explained in section 2.1.1, the study was done with no prior information of the company. As a result, questions requesting sensitive data that did not preserve the anonymity of the respondent were included in the surveys. This information was crucial for the study and could not be taken out. It was necessary to know what the distribution of the development team was and what the location and product of the PdMs were.

   As it was explained in section 2.2.2.3, the PdMs questionnaire did not include questions requesting sensitive data in the first version. However, after the pilot study the author of this thesis realized that including this information was necessary. As a result, the questionnaire was later modified requesting sensitive data that may have lowered the response rate. The main purpose of these surveys was to detect the current and the desired agility of developers and PdMs. Hence, the main purpose was still achieved and the inclusion of sensitive data was not a major threat in the study.

2. **Generalization of the study**

   The number of respondents in both surveys was low. This could threaten both the internal and external generalization of the study. The author is not trying to claim that the findings of this study are generalizable for the whole software industry.

   The aims of this study covered determining if a global software development company was prepared to be agile, studying hindrances for agility, and providing recommendations to overcome the detected hindrances.

   As a result, this study makes impact in one company. However, due to the low number of respondents in the surveys, there is a chance that some hindrances could not be identified.

3. **Roles selection**

   The organizational mapping provided by the company showed seven different roles. Due to time constraints and the scope of the study, only three roles were selected for analysis. The roles not covered in the study are Project Manager, System Architect, Project Sponsor and Senior Manager. As a result, there is a chance that some information coming from these last roles was not elicited and, some hindrances were not detected.

4. **Suitability of the Selected Literature for Recommendations**

   There is a threat in the selected systematic review for recommendations. As it was explained in section 4.1 several systematic reviews were analyzed and evaluated but only one of them was selected for the study. This systematic review had some constraints in terms of publication year and covered databases. These constraints were studied (see section 4.1) and the impact they could have on the study was determined. An extension of this systematic review was not considered strictly necessary.

   One of the main purposes of this study was to provide recommendation on how to overcome the detected hindrances. Hence, this purpose was still achieved and these

constraints were not a major threat in the study. However, the impact of these constraints will grow as time goes by and could pose a major threat to the study in the future.

5. Converted Data

The respondents of the surveys were asked to choose or write open answers about the frequency of meetings with product managers and team mates. When analyzing this data the author realized that it would provide a much more useful information if it was shown as quantitative instead of qualitative. As a result, a conversion from qualitative to quantitative was done by assigning a frequency frame to each point of the likert scale.

The main purpose of this converted data was to determine what the frequency of intra-team and developers-product managers meetings was. The purpose was still achieved, so the nature of this data was no longer a major threat.

6. Interviewee Misgiving

While designing the survey for the developer, it was not taken into account that the information that was requested to be revealed was, from the point of view of the developer, a risk. In other words, while performing the interview, the interviewer realized that the interviewee was trying to hide his lack of understanding of the development process by responding vague answers. This could have posed a threat in the detection of the root causes of the hindrances for agility found in the surveys. However, the interview was stopped to tranquilize the developer and inform him that the results were going to be used for suggesting improvements with no consequences for him. After this pause, the developer seemed more comfortable and answered frankly, revealing the information he was asked for.

The main purpose of the surveys was to discover and understand the root causes of the detected hindrances. This purpose was achieved so the misgiving of the developer was no longer a threat to this thesis.

7 CONCLUSION

7.1 Revisiting Research Questions

7.1.1 R.Q. 1

Two surveys were conducted to determine if the company had agile features and to detect the challenges for agile in the development process. They had the purpose of determining the current agility and agile willingness of two roles involved in global distributed projects: developers and product managers. Guidelines suggested by Kitchenham [35] were followed to design both surveys. The questions included in each of the surveys are explained and documented in sections 2.1.2.1 and 2.2.2.1.

The questionnaires were completely filled by 15 and 4 developers and product managers respectively. Both kinds of subjects were detected to have some agile features and to mostly prefer the agile principles over the traditional ones. This does not mean that the whole population would like to be more agile. However, it shows that there is a favorable trend for adopting agile.

Through the analysis of the retrieved data it was determined that the company was prepared to adopt agile. Several challenges for agility were discovered: process visibility, roles blurriness, documentation overproduction and lack of product manager's feedback integration. Without the mitigation of these challenges, the company may not be able to achieve a higher level of agility in the development process' execution. Hence, a root cause analysis was performed to determine the hindrances of those challenges.

7.1.2 R.Q. 2

As aforementioned, some challenges for agility were detected through the conduction of two surveys. To detect the hindrances of those challenges, a root cause analysis was performed. The Root Cause Analysis was interview based and was designed in an incremental way, i.e. the interview questions were designed looking at the results retrieved in the previous research step, following the guidelines of [43-45].

Two interviews were conducted. The two interviewees were involved in global distributed environments, playing two different roles in two different locations: a unit manager from Sweden and a developer from Russia.

The excess of documentation was reported necessary for the maintenance of old products. The heavy weight documentation was reported necessary to obtain a CMMI level 3 certification. The root causes were detected to be:

- **Process Visibility**: An abstraction of the developer over the development process.
- **Lack of Product Manager's Feedback Integration**: Lack of pro-activity from the side of the Product Manager and dissatisfaction with the feedback management. The Product Managers also lack visibility of the product progress.
- **Roles Blurriness**: Poor definition of roles and/or loss of roles definition and responsibilities due to changes in the process and team configuration.

Hence, recommendations based on literature to overcome those root causes were provided.

7.1.3 R.Q. 3

To mitigate or overcome the hindrances for agility detected in section 3, a set of recommendations inspired in literature were provided. There was no necessity for a systematic

review since there was one that exactly fitted the needs of this thesis [24]. An evaluation of the suitability of this systematic review was performed following Kitchenham's guidelines [67]. Finally, the systematic review was considered suitable.

Several recommendations inspired on the tools and best practices recorded in [24] were provided. Recommendations for a better roles definition and circulation are explained in section 4.2, recommendations to improve the product manager's proactivity to get an early and continuous feedback are depicted in section 4.3 and recommendations to spread a better understanding of the development process among the staff of the company are provided in section 4.4.

The study was evaluated with a survey to the representative of the company. The work was reported not to be novel and the solutions were reported to be applied or considered to be applied. As a result, they are useful.

7.2 Future Work

The results of this study could be used as foundation for further investigation in the area of GSD, specifically in the part related to challenges in Distributed Agile Development. The author of this thesis has identified some probable future studies:

1. **Extension of the Systematic Review**

   The systematic review used in section 4 could be extended. In this way the collection of available tools, models and best practices to address the hindrances for agility found in this case study could be enlarged.

2. **Repeat the study with a wider population**

   For future works it is also important to repeat the study with more subjects. This could increment the extrapolation factor and provide more reliability to the collected data. More hindrances for agility could be found and they could be further studied finding their root causes.

3. **Study teams perspective**

   Each team has a different perspective and a different way of working. This thesis has been focused on contrasting perspectives in terms of roles. For future works the detection of hindrances focused in terms of teams could also be an option.

4. **Implementation of the Proposed Recommendations**

   A study on how to implement the proposed recommendations without affecting the productivity of the company could continue this work. This could open several research paths.

8 AFTERWORD: THESIS' CHALLENGES AND EVOLUTION

There have been different challenges that the author of this thesis had to overcome. The main aims of the thesis proposal were to determine the suitability of different software life cycle models in GSD and to investigate the availability of different toolkits suitable for a successful GSD. This topic has evolved from the original until the current one through different phases.

Working in a thesis with a company has been very interesting. However, it had its advantages and disadvantages. The thesis does not only depend on the author, but also on the company proactivity. This increases the risk of the thesis to be compromised.

In this case study, the master thesis supervisor has been the communication bridge between the master thesis student and the company responsible. Each time the author required some information from the company two emails had to be sent (student-supervisor, supervisor-responsible) and two emails had to be received (responsible-supervisor, supervisor-student). This reduces the communication speed and can generate gaps in the work process. However, the student and the supervisor had coordinated good to overlap phases while the student was waiting for the required information to be sent.

The analysis of the two surveys has been also very challenging. The selected population did not respond as quickly as expected and there was a long elapsed time while waiting for the answers. It took more than 3 weeks per each survey to be completed with an acceptable number of respondents. One reminder had to be sent per each survey to get more respondents.

The first survey had 15 respondents. Data coming from this survey was considered at first to be analyzed by teams. The developers were grouped in this way to determine the perspectives from the point of view of the team. However, 4 respondents were unable to be classified in any team because of inconsistencies in the replies. These subjects were discarded in this first version of the analysis. A study focused on descriptive statistics was performed. Conclusions were drawn over this analysis.

The population was not selected by the author of this thesis but by the company. In the second survey, the author realized that the product managers selected for study did not match the products developed by the teams of the first survey. As a result, the analysis of the first survey was changed from teams to roles, and from descriptive statistics to scatter plots analysis.

The second survey was analyzed also in terms of individuals, also focused on roles. The responsible of the company selected some products that were not developed in a distributed way. As a result this data had to be dropped and the analysis had to be done with half of the data.

When analyzing the data from both surveys, the author of the thesis detected that there could be overproduction in documentation. This could be a source of waste in the development process. As a result, the author made a proposal to apply a Lean development tool known as Value Stream Mapping (see Appendix I - VSM Proposal – A Tool for Eliminating Waste and Enhance Value). The purpose of this tool was to determine sources of waste and mitigate them.

A 3 days workshop was proposed. Besides, a project manager, a representative of the developers and a product manager from each development process family to be studied were required to attend. However, the representative only accepted a 2 hours meeting just with him. This was not enough to successfully apply Value Stream Mapping. As a result, the focus of the interview was changed from Value Stream Mapping to Root Cause Analysis. The target of the interview was then to detect the root causes of the detected challenges in the development process.

After having the first interview, the whole thesis focus was readjusted by the company responsible, the thesis supervisor and the master thesis student. The new focus was to study the hindrances for agility in the software development process. For this new focus, specific material from the company was requested. Some information arrived three weeks after the request. The thesis had to be readjusted according to this new information. Some other information was sent a month after the request and was not considered for the thesis since it was out of scope and time.

The general idea of the thesis was to determine if the company was prepared to be agile and what the hindrances for agility were. The experience of the author is that none of the roles involved in this study demonstrated agility towards the thesis. In fact, the agility of the author was constrained by the agility of the company respondents and responsible. However, it was quite interesting to interact with a real environment; it made both the thesis and the author full of firsthand knowledge about challenges in Global Software Development Environments.

9 APPENDIX

9.1 Appendix I - VSM Proposal – A Tool for Eliminating Waste and Enhance Value

In the light of previous findings from developers' survey in Emerson in Sweden and Russia, we would like to propose continuation of understanding the suitability of process lifecycle through Value Stream Mapping. In particular, our observations suggest that there is an incomplete understanding of the development process among the developers and a lack of flexibility in certain processes.

Overview: Value Stream Mapping (VSM) is a set of specific actions that helps in analyzing process flows and thus facilitates the critical management tasks. This requires drawing maps (value stream map) of the current and the desired development process flow. After that, the planning and implementation of the necessary detected changes is agreed (see Figure 20a). VSM could serve as tool for managing the flow of the development process, eliminating the waste, enhancing value and suggesting changes to move from the current to the desired development process model (see Figure 20b).

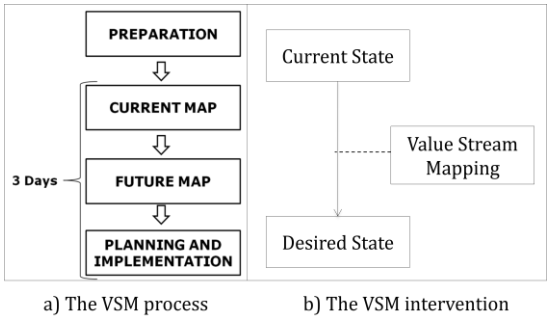

Figure 20. VSM process and intervention

Expected results: One of the expected benefits of VSM is the help it provides to redefine, document and clarify the development process, so that it becomes more transparent and easier to understand.

Participants: VSM workshop participants shall include people involved in the development of the family of products or processes selected for analysis. We propose to select one family of products (or projects) following the same process model and invite a product owner, a project manager and two representatives of the development team (ideally one from Sweden and one from Russia). This will potentially require the use of videoconference and, if possible, a digital whiteboard or similar tools.

Procedure: VSM shall be applied in a three days workshop (3 hours per day). The workshop requires preparation, in which the value stream managers (the master thesis student and supervisor) and Darel Cullen decide in advance who will participate in the mapping team and which family of products or projects will be studied. The VSM workshop then comprises of three activities (see

Table 30).

Table 30. Schedule for the VMS workshop

Date and Time	***Activity***	***Explanation***
April 11, 9:00-12:00	*Current State*	*Agreement on the map of the current state*
April 12, 9:00-12:00	*Future State*	*Agreement on map of the future desired state*
April 13, 9:00-12:00	*Planning & Implementation*	*Development of a plan to achieve the future state*

10 REFERENCES

[1] S. Sahay, B. Nicholson, and S. Krishna, *Global IT Outsourcing: Software Development Across Borders*, Cambridge University Press, 2003.

[2] D. Damian and D. Moitra, "Guest Editors' Introduction: Global Software Development: How Far Have We Come?," *IEEE Software*, vol. 23, Sep. 2006, pp. 17-19.

[3] E. Conchuir, H. Holmstrom, P. Agerfalk, and B. Fitzgerald, "Exploring the Assumed Benefits of Global Software Development," *1st International Conference on Global Software Engineering (ICGSE'06)*, IEEE, 2006, pp. 159-168.

[4] J. Kontio, M. Hoglund, J. Ryden, and P. Abrahamsson, "Managing commitments and risks: challenges in distributed agile development," *26th International Conference on Software Engineering*, IEEE Computer Society, 2004, pp. 732-733.

[5] E. Carmel, *Global software teams: collaborating across borders and time zones*, Prentice Hall PTR, 1999.

[6] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," *Empirical Software Engineering*, vol. 15, 2009, pp. 91-118.

[7] D.W. Karolak, *Global Software Development: Managing Virtual Teams and Environments*, Los Alamitos, CA, USA: IEEE Computer Society, 1998.

[8] P.J. Ågerfalk, B. Fitzgerald, B. Lings, B. Lundell, and E.Ó. Conchúir, "A Framework for Considering Opportunities and Threats in Distributed Software Development," *Proceedings of the International Workshop on Distributed Software Engineering*, 2005, pp. 47-61.

[9] J. Kotlarsky and I. Oshri, "Social ties, knowledge sharing and successful collaboration in globally distributed system development projects," *European Journal of Information Systems*, vol. 14, 2005, pp. 37-48.

[10] H. Holmstrom, E. Conchuir, P. Agerfalk, and B. Fitzgerald, "Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance," *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*, IEEE, 2006, pp. 3-11.

[11] K.E. Nidiffer and D. Dolan, "Evolving Distributed Project Management," *IEEE Software*, vol. 22, 2005, pp. 63-72.

[12] T. Mcbride, "The use of project management mechanisms in software development and their relationship to organizational distance : An empirical investigation," Faculty of Information Technology University of Technology, Sydney, 2005.

[13] K. Naik and S. Kishore, *Software Requirements And Estimation*, India: Tata McGraw-Hill, 2001.

[14] S. McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006.

[15] M. Fowler and J. Highsmith, "The agile manifesto," *Software Development Magazine*, vol. 9, 2001, p. 28–35.

[16] M. Laanti, "Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation," *Information and software technology*, vol. In Press,, 2010.

[17] M. Paasivaara and C. Lassenius, *Could Global Software Development Benefit from Agile Methods?*, IEEE, 2006.

[18] K. Sureshchandra and J. Shrinivasavadhani, "Adopting Agile in Distributed Development," *2008 IEEE International Conference on Global Software Engineering*, Aug. 2008, pp. 217-221.

[19] P. Xu, L.A.N. Cao, and K. Mohan, "Can distributed software development be agile?," *October*, vol. 49, pp. 41-46.

[20] M. Kajko-Mattsson, G. Azizyan, and M.K. Magarian, "Classes of Distributed Agile Development Problems," *2010 Agile Conference*, Aug. 2010, pp. 51-58.

[21] B.S. Drummond and J.F. "Jf" Unson, "Yahoo! Distributed Agile: Notes from the World Over," *Agile 2008 Conference*, Aug. 2008, pp. 315-321.

[22] M. Paasivaara, S. Durasiewicz, and C. Lassenius, "Distributed Agile Development: Using Scrum in a Large Project," *2008 IEEE International Conference on Global Software Engineering*, Aug. 2008, pp. 87-95.

[23] P.J. Ågerfalk and B. Fitzgerald, "Flexible and Distributed Software Processes: Old Petunias in New Bowls?," *Communications of the ACM*, vol. 49, 2006, pp. 26-34.

[24] F.Q.B. Da Silva, C. Costa, A.C.C. Franca, and R. Prikladinicki, "Challenges and Solutions in Distributed Software Development Project Management: A Systematic Literature Review," *2010 5th IEEE International Conference on Global Software Engineering*, 2010, pp. 87-96.

[25] M. Paasivaara, S. Durasiewicz, and C. Lassenius, "Using Scrum in Distributed Agile Development: A Multiple Case Study," *2009 Fourth IEEE International Conference on Global Software Engineering*, Jul. 2009, pp. 195-204.
[26] M. Cohn, *Succeeding with Agile: Software Development Using Scrum*, Addison-Wesley, 2009.
[27] D. Šmite, "R2D2 - Decission-Support for Offshoring Software Development, http://web.me.com/darja.smite/R2D2 , 2010/09/19 [2010/02/03]."
[28] F. Pettersson, M. Ivarsson, T. Gorschek, and P. Öhman, "A practitioner's guide to light weight software process assessment and improvement planning," *Journal of Systems and Software*, vol. 81, 2008, pp. 972-995.
[29] T. Gorschek, "Requirements engineering supporting technical product management," *Engineering*, 2006.
[30] V.R. Basili, "Quantitative evaluation of software methodology," 1985.
[31] C.P. Team, "CMMI ® for Development , Version 1 . 2," *Framework*, 2006.
[32] T.P. Rout, *ISO/IEC 15504 and Spice*, John Wiley & Sons, Inc., 2002.
[33] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, Dec. 2009, pp. 131-164.
[34] B.A. Kitchenham and S.L. Pfleeger, "Principles of Survey Research Part 1: Turning Lemons into Lemonade," *Software Engineering Notes*, vol. 26, 2001, pp. 16-18.
[35] B. Kitchenham and S.L. Pfleeger, "Principles of Survey Research," *Software Engineering Notes*, vol. 27, 2002, pp. 16-24.
[36] Google, "Google Docs Form Creation Tool," 2011.
[37] IBM, "IBM SPSS Statistics," 2011.
[38] Microsoft, "Microsoft Excel," 2011.
[39] R. Prikladnicki and J.L.N. Audy, "Process models in the practice of distributed software development: A systematic review of the literature," *Information and Software Technology*, vol. 52, 2010, pp. 779-791.
[40] C. Kussmaul, *Outsourcing and offshoring with agility: a case study*, Springer Berlin / Heidelberg, 2004.
[41] M. Fowler and J. Highsmith, "The agile manifesto," *Software Development*, vol. 9, 2001, p. 28–35.
[42] F. Lanubile, C. Ebert, R. Prikladnicki, and A. Vizcaíno, "Collaboration Tools for Global Software Engineering," *IEEE Software*, vol. 27, 2010, pp. 52-55.
[43] J.J. Rooney and L.N.V. Heuvel, "Root Cause Analysis For Beginners," *Quality Progress*, 2004, pp. 45-53.
[44] R.J. Latino and K.C. Latino, *Root Cause Analysis - Improving Performance for Bottom-Line Results*, New York, New York, USA: Taylor & Francis Group, LLC, 2006.
[45] M. Ammerman, *The root cause analysis handbook. A simplified approach to identifying, correcting, and reporting workplace errors. 2*, Productivity Press, 1998.
[46] B. Kitchenham, "Procedures for Performing Systematic Reviews," *Keele UK Keele University*, vol. 33, 2004, p. 28.
[47] T. Chow and D. Cao, "A survey study of critical success factors in agile software projects," *Journal of Systems and Software*, vol. 81, 2008, pp. 961-971.
[48] F.D. Report, "3rd Annual Survey : 2008 'The State of Agile Development'," *Time*, 2008.
[49] B. Kitchenham and S.L. Pfleeger, "Principles of Survey Research Part 4: Questionnaire Evaluation," *Software Engineering Notes*, vol. 27, 2002, pp. 20-23.
[50] G. Iarossi, *The Power of Survey Design: A User's Guide for Managing Surveys, Interpreting Results, and Influencing* , World Bank Publications, 2006.
[51] Facebook ©, "Facebook," 2011.
[52] LinkedIn ©, "LinkedIn," 2011.
[53] "DSDM Consortium - Enabling Business Agility."
[54] C. Young and H. Terashima, "How Did We Adapt Agile Processes to Our Distributed Development?," *Agile 2008 Conference*, Aug. 2008, pp. 304-309.
[55] J. Evaristo, "A dimensional analysis of geographically distributed project teams: a case study," *Journal of Engineering and Technology Management*, vol. 21, 2004, pp. 175-189.
[56] N.K. Denzin and Y. Lincoln, *The sage handbook of qualitative research. Third edition*, Sage Publications, 2005.
[57] M.B. Miles and A.M. Huberman, *Qualitative Data Analysis*, Sage, 1994.

[58] K.S. Khan, G. Ter Riet, J. Glanville, A.J. Sowden, and J. Kleijnen, "Undertaking Systematic Reviews of Research on Effectiveness CRD's Guidance for those Carrying Out or Commissioning Reviews," vol. 4, 2001, p. 152.

[59] V. Casey and I. Richardson, "Project Management Within Virtual Software Teams," *1st International Conference on Global Software Engineering (ICGSE'06)*, Ieee, 2006, pp. 33-42.

[60] M. Leszak and M. Meier, "Successful Global Development of a Large-scale Embedded Telecommunications Product," *2nd International Conference on Global Software Engineering (ICGSE'07)*, 2007, pp. 23-32.

[61] N. Mullick, M. Bass, Z. Houda, P. Paulish, and M. Cataldo, "Siemens Global Studio Project: Experiences Adopting an Integrated GSD Infrastructure," *2nd International Conference on Global Software Engineering (ICGSE'06)*, 2006, pp. 203-212.

[62] J. Ralyte, X. Lamielle, N. Arni-Bloch, and M. Leonard, "A framework for supporting management in distributed information systems development," *Research Challenges in Information Science 2008 RCIS 2008 Second International Conference on*, 2008, pp. 381-392.

[63] M. Paasivaara, S. Durasiewicz, and C. Lassenius, *Distributed Agile Development: Using Scrum in a Large Project*, IEEE, 2008.

[64] B.C.Y. Tan, H.J. Smith, M. Keil, and R. Montealegre, "Reporting Bad News About Software Projects : Impact of Organizational Climate and Information Asymmetry in an Individualistic and a Collectivistic Culture," *Engineering*, vol. 50, 2003, pp. 64-77.

[65] G. Hofstede and G.J. Hofstede, *Cultures and Organizations: Software of the Mind*, McGraw-Hill, 2005.

[66] P.S. Brockmann and T. Thaumuller, "Cultural Aspects of Global Requirements Engineering: An Empirical Chinese-German Case Study," *2009 Fourth IEEE International Conference on Global Software Engineering*, 2009, pp. 353-357.

[67] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," *Engineering*, vol. 2, 2007.