Investigating Communication with Stakeholders in Agile Requirements Engineering

Master's thesis in

MOHAMMED MEGDAD

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2017

Master's thesis 2017:NN

Investigating Communication with Stakeholders in Agile Requirements Engineering

MOHAMMED MEGDAD

Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2017

Investigating Communication with Stakeholders in Agile Requirements Engineering
MOHAMMED MEGDAD

Supervisor: GRISCHA LIEBEL , Department of Computer Science and Engineering
Examiner: ROBERT FELDT, Department of Computer Science and Engineering

Master's Thesis 2017:NN
Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology
University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2017

Investigating Communication with Stakeholders in Agile Requirements Engineering
MOHAMMED MEGDAD
Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg

Abstract

Background: It is unclear how Requirement Engineering (RE) and Agile Software Development (ASD) can work together. Several studies have investigated this phenomenon, and their results show several challenges in using RE and ASD, such as communication gaps. However, there is a scarcity of publications that provide empirical data on the combination of RE and ASD.

Objective: The purpose of this research is to investigate the challenges of using RE-related activities in ASD and how these challenges affect the communication with stakeholders.

Method: A case study method is used to investigate the IT department in a large organization, the Swedish public employment agency Arbetsförmedlingen (AF). Data were collected through twelve semi-structured interviews, telephone interviews, and email contact. Additionally, tools such as Jira was used to investigate the quality of requirements whereas Usabilla was accessed to examine the feedback received from external stakeholders.

Results: Several challenges from implementing RE-related activities in ASD were discovered such as clarity of requirements and using customer representatives. Some of these challenges relate to a purely agile context and are impacted by a lack of adoption of agile practices. Others are more generic for RE and are likely to occur also in non-agile environments. The discovered challenges have a negative impact on the communication with different stakeholders, which negatively affects the development process.

Conclusion: The results indicate that physical communication with customers plays a significant role in the quality of the delivered products, but the customer representatives are unable to play the role of the customer in the development process. Furthermore, our results show the importance of understanding agile practices on all levels, including the management and customers.

Keywords: Requirements, Agile, Communication, Stakeholders, Customers, Challenges

Acknowledgements

Mohammed Megdad, Gothenburg, 05 2017

I would like to thank various individuals for their guidance and support throughout the conduct of this thesis. These include:

My academic supervisor, Grischa Liebel, for guidance, ensuring a critical review of the work, and for providing the resources needed to complete this research,

My company contact, Jens Eriksson, for collaboration, guidance and support throughout the study,

My manager in Arbetsförmedlingen, Patricia Anselmo, for support and facilitate my work schedule,

My wife, Noor, for standing up for me during the difficult and stressful moments,and

My mother, father, brothers and sisters for raising me up to become who I am today. I did it after 11 years.

I dedicate this work to my grandfather, for his prayers, endless love, support, and encouragement.

Contents

1
Introduction

Communication is a very significant factor in software development. However, the communication between customers and software companies is still a challenge [1]. One of the difficulties that affect the communication is the ineffective way of requirements documentation, which always needs to be updated. ASD has been proposed to improve the communication through providing methods and practices and suggests effective practices for better collaboration among all stakeholders, developers, and managers [2], [1].

ASD has gained a large popularity in software engineering practice [3], [4]. Many companies and organizations started to use ASD instead of traditional development methods because it improves the ability to create more efficient and effective communication between the companies and their customers [1]. Movement from traditional development to ASD needs assistance. This assistance provides the managers in the organizations with an evolutionary path to increase the software maturity in stages [5]. Several models provide assistance for this movement such as Agile Maturity Model (AMM). This model provides guidance and control over software development through selecting development strategies and identifying the critical issues related to software quality [5].
Several studies argue that using ASD has a positive effect on the communication between organizations and their customers through improving both formal and informal communication [1]. ASD aims at decreasing documentation and leading to a more productive software development. However, other studies show that the companies which use ASD suffer from increasing the gap in communication between stakeholders [1]. This communication gap might lead to a complete failure in the software development process [1]. Also, using RE in ASD can cause a gap in communication and over scoping, specifically in large software companies [2].

RE is one of the most important phases of software development. In this phase, the requirements for software products are gathered, analyzed, documented, and managed throughout the SE lifecycle [6]. Traditional RE includes different activities such as elicitation, analysis, specification, verification and validation of requirements [7], [6]. In agile development, RE-related activities are not explicit. For example, user stories ("A user story is a description of a feature that provides business value to the customer" [8]) are used instead of elicitation and frequent review meetings and acceptance tests for requirements validation. However, many projects fail because of different factors which are related to RE. These factors include, but are not limited to incomplete requirements or useless requirements [6]. According to Ramesh, Cao, and Baskerville [9], there are two risks identified from using RE in ASD, which are customer inability to understand the impact of agile development and neglecting non-functional requirements. Moreover, there are many other problems identified such as "the use of customer representatives, minimal documentation, inappropriate architecture, growing technical debt, tacit requirements knowledge, and imprecise effort estimation" [10], [2].

The purpose of this research is to investigate the challenges of using RE-related activities in ASD and how those challenges could affect the communication with stakeholders. To do so, we use a case study method, studying the IT department of one large organization, AF, that recently tried to transition to ASD, but now encounters several problems in relation to requirements. Our results show that these problems relate to several different aspects of development: While some problems indeed arise through the combination of ASD and RE-related activities, others relate to a lack of adoption of agile practices or are not related to ASD at all.

The remainder of this report is organized as follows. Chapter 2 presents background and related work on agile RE, communication in software development and agile methods and practices. In Chapter 3, we present our case organization; the IT department and their stakeholders, agile practices and RE in the department and which tools are used in the department. Chapter 4 presents our research questions and hypotheses, our research methodology and why we used the case study approach and threats to validity. In Chapter 5, we present the research finding. The discussion will be presented in Chapter 6, while conclusion is stated in Chapter 7.

2
Background

In this chapter, we present the related work and background about RE in ASD and how the challenges of using this phenomenon affect the communication with stakeholders.

2.1 Traditional Requirements Engineering

Chemuturi [11] defines RE as " a process of converting the specifications of customers into such artifacts that are used by artisans to produce the product that fulfills the customer specifications." The main goal of RE is to produce a complete and consistent set of requirements that reflects the customer's need [12]. Moreover, RE helps the development teams to plan their development process to prevent costly rework [13]. According to Sillitti and Succi [6], RE has common activities which are elicitation, analysis, specification, validation, prioritization of requirements and system validation.

Heikkilä et al. [3], defined the RE activities. In the first activity, requirement elicitation, initial information regarding requirements are gathered. These information is analyzed in the second activity in order to create a common understanding of the requirements. In third activity, the understanding of requirements turned into system specifications. Requirement validation is the fourth activity which supports the previous activities by identifying and correcting errors in the requirements. In the fifth activity, requirements prioritization and valuable requirements are identified. In the last activity, system validation, the developed product is validated by comparing it to the requirements.

2.2 Agile Requirements Engineering

It is unclear how RE and ASD work together. According to Heikkilä et al. [10], the definition of agile RE is difficult since agile methods such as Scrum includes RE activities in the overall method description. Based on that, it is unclear the difference between agile RE and traditional RE.

RE in ASD is based on knowledge of individuals rather than process orientation [10]. According to Paetsch, Eberlein, & Maurer [8], RE activities are present in all agile processes. However, these activities are not explicit and could be integrated with other activities. For example, the Planning Game in Extreme Programming is an elicitation and analysis activities.

Using RE in ASD comes with several challenges. One of those challenges is the communication gap and overscoping in large software companies [2]. According to Ramesh, Cao, and Baskerville [4], there are two risks identified from using RE in ASD, which are customer inability to understand the impact of agile development and neglecting non-functional requirements. Moreover, there are many other problems identified such as "the use of customer representatives, the user story requirements format, the prioritization of requirements, growing technical debt, tacit requirements knowledge, and imprecise effort estimation" [10],[2].

Ramesh, Cao, and Baskerville also identified several challenges from using RE in ASD such as "overemphasis of functional requirements, inadequate inspection of requirements and incorporation of designs in requirements" [4].

Heikkilä et al. [10], mentioned that the definition of combining RE and ASD is still weak. However, they defined in a mapping study the problematic areas of agile RE, namely, using customer representatives, the user story requirements format, the prioritization of requirements, growing technical debt, tacit requirements knowledge, and imprecise effort estimation [10].

Another case study by Bjarnason, Wnuk, and Regnell, which studied the transition of large software companies from traditional development into ASD shows that implementing RE in some ASD practices faces several challenges, such as communication gaps [14]. Moreover, Turk, France, and Rumpe studied in their paper [15] the limitations of ASD in the case of involving large teams and developing large software, which reduce effectiveness of some agile practices such as face-to-face communication. Sillitti and Succi in their book [6] argue that customer satisfaction comes from implementing ASD correctly.

Inayat et al. [13], identified several agile RE challenges and their proposed solution. These are shortly outlined in Table 2.1.

Agile RE challenges	**Proposed solutions**
Neglecting NFRs	Methods to tackle NFRs in agile methods
Lack of requirements traceability	Methods to enable adequate requirements traceability
Incorrect requirements prioritization	Methods for value based requirements prioritization
Minimal requirements documentation	Collocated teams, Face-to-face communication and On-site customer
Contractual issues	Legal measures and Fixed price contracts
Customer availability	Proxy customers
Customer agreement	Appointing appropriate customer representative

Table 2.1: Agile RE challenges and their proposed solutions [13]

2.3 Communication in software development

Communication is a way to manage the coordination process in an organization. It is an essential success factor for software development. However, other studies show that communication is a key challenge in software development as well as distributed software projects. This challenge hinders establishing a good communication between customers and software companies [16]. Macualary has explained specifically that insufficient communication between people involved in RE process and lack of knowledge about what the system should do, impede the understanding of requirements. This results in poorly delivered systems [17]. The provided solution for communication problems in software companies is "agile principles and methods", which suggests better practices for collaboration between all stakeholders, developers and qualified managers [2], [1], [18].

Goles has defined two different types of communications which are formal and informal communications [19]. Pikkarainen in his article [1] argues that agile practices improve both formal and informal communication and have positive effects on communication among teams. Sprint planning, open office space and daily meeting are suggested as good practices to communicate requirements in agile teams and outside the organization through facilitating dependencies between development teams and stakeholders [1].

2.3.1 Formal Communication

Formal communication refers to explicit communication such as groups, weekly or steering meetings. It can be training sessions which distribute knowledge but may inhibit creativity [19], [20]. Formal communication is central to any well-developed field and provides newly discovered information as well as suggestions for successful completion of tasks [21]. Formal communication has different channels where the information and instructions are passed downward and upward along these channels [22]. This type of communication follows the official hierarchy of the organization. Different studies have proposed that formal communication has a positive effect on productivity, satisfaction, and trust. Moreover, it reduces conflict and coordination cost [22].

2.3.2 Informal Communication

Informal communication refers to face to face communication between co-located teams, unscheduled meetings, or coffee break conversations [20]. It could also be in form of telephone calls, mail, video, and audio. This type of communication is the best way to build trust and satisfaction between team members, through talking to the employees about their problems, attitude, and other things. Moreover, informal communication is a source of information for the managers, that helps them to lead the employees successfully [22].

Agile practices proposed to increase the informal communication in the organizations through face to face conversations and daily work together [19]. Gilsdorf [23] noted that informal communication covers the gap in communication that formal communication fails to address. The gap in communication comes either from insufficient information or ambiguity of information systems. This lack of formal communication increases people's reliance on informal communication [24]. However, this type of communication may involve certain amounts of knowledge atrophy which hinders passing knowledge between people. Therefore, passing accurate knowledge between people is not guaranteed [20]. Moreover, other studies indicate that using informal communication results in higher defect rates [22].

2.4 Agile Methods and Practices

To be agile means that development teams deliver quickly, change quickly and frequently [2]. Agile techniques vary in practice but share the same characteristics such as iterative development and a strong emphasis on communication [4]. According to Cohen [4], there are different methods and practices such as Extreme Programming, Scrum, Kanban, Crystal Methods, Feature Driven Development, Lean Development, and Dynamic Systems Development Methodology. In the next two sections, we explain two different practices which are implemented in the IT Department in AF.

2.4.1 Scrum

Scrum is one of the more widely used agile practices [25]. Schwaber has defined Scrum as an enhancement of commonly used iterative and incremental object-oriented development cycles. It consists of three different roles, events, and artifacts. Figure 2.1 summarizes the process.

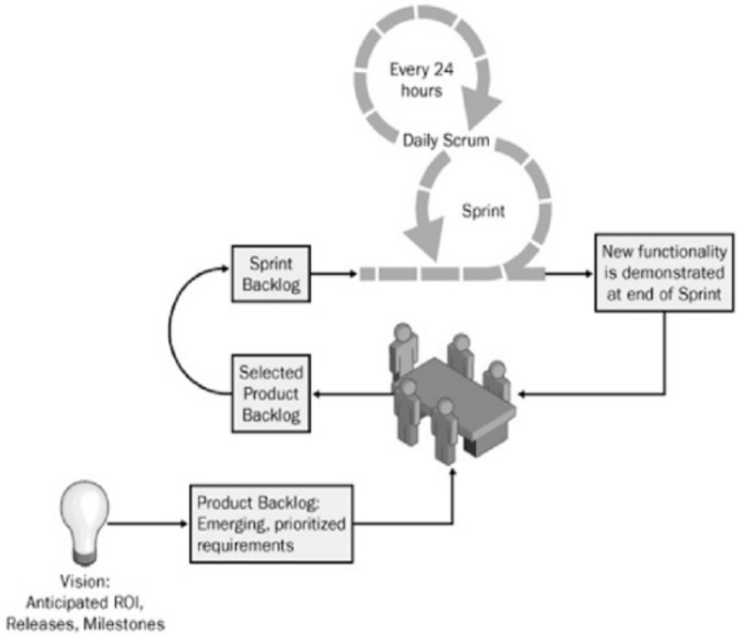
Every 24 hours
Daily Scrum
Sprint
Sprint Backlog
New functionality is demonstrated at end of Sprint
Selected Product Backlog
Product Backlog: Emerging, prioritized requirements
Vision: Anticipated ROI, Releases, Milestones

Figure 2.1: Scrum Process Overview [26]

In Scrum, there are three defined roles. The first role is Product Owner (PO) who is responsible for eliciting requirements. The second role is Scrum Master who is responsible for teaching and spreading Scrum knowledge through the team. The last role is development team which according to Cohen, consists of seven people to the utmost, which includes at least one developer, quality assurance engineer and a documenter [25].
Scrum has different events such as a Sprint. During the sprint, the team members choose the tasks from their sprint backlog. At this time, all tasks in this sprint are frozen until the sprint duration is completed. It is very important to have a short daily meeting (Scrum meeting) to identify problems, keep the focus on the goal and enhance the communication inside the team and with the customers [4]. A post-sprint meeting is held to analyze and demonstrate the current system.
Three different artifacts are defined in Scrum. The first one is the Product Backlog, which contains all the events, features and requirements that need to be done by a development team The second one is Spring Backlog, which contains the prioritized features and events for the next sprint. Thirdly, the Burndown chart is an optional artifact in Scrum. This chart shows the cumulative work remaining [4].

Scrum has different advantages in the development process compered to traditional development methodologies. One of these advantages is providing a mechanism of planning for product release. This enables organizations to deliver the most value to the customer and doing changes on the project at any time during development [25]. Another advantage is that the collaboration between small teams increases their knowledge about the development process. Moreover, Scrum increases the responsibility among team members, where every person is responsible for his work [25]. However, adopting Scrum can be challenging. Marchenko and Abrahamsson define several challenges related to Scrum practices which are: lack of clear management, fitting scrum, and short iterations, over-individualism where some team members still prefer individual work, too much committing during the sprint and difficulty in tracking process [27].

2.4.2 Kanban

Kanban is a manufacturing scheduling system [28], which received its name from the Japanese term for "Signboard". The key point of using Kanban is the absence of obligatory iterations. In this methodology, project teams try to visualize the workflow, limit work in progress (WIP) at each workflow stage through developing those items which are requested, and measure cycle time. The Kanban board provides visibility of software process through showing what developers work on. This leads to a constant flow of release work and aims for a quick adoption of the software development through using shorter feedback loops [28].
Comparing to traditional development methodologies, many advantages have been identified through using the Kanban methodology. These include: simplicity, focus on workflow and no obligatory iterations [29]. According to Anderson [30], Kanban limits work in progress, which leads to a sustainable development process through visualizing process problems and decreasing defects. This builds trust among cus-

tomers through having high-quality products and high team performance.
Figure 2.2 shows an example of a Kanban board.

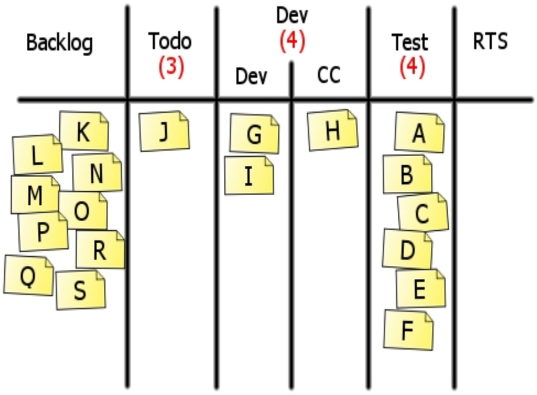

Figure 2.2: A Kanban board (from http://brodzinski.com/2009/11/kanban-story-kanban-board.html)

2.5 Maturity Models

Organizations that are moving from traditional to agile software development need assistance to achieve this transition in order to improve agile capability [31]. Several models exist that assess the maturity of software development processes. These models can help organizations to understand where and what to improve. Additionally, they can help to understand external assessors where problems are lying. Here, we introduce to maturity models. First, the Agile Maturity Model (AMM) is targeted at assessing the maturity of agile processes. Secondly, the Capability Maturity Model (CMM) targets process maturity in general. In the context of this study, the two models can help us to understand how mature the software development processes are, both with respect to agile adoption and to traditional software development

2.5.1 Agile Maturity Model

Agile Maturity Model (AMM) is developed to help and guide organizations in agile adoption and enhance agile principles such as customer satisfaction and software quality [31], [32]. AMM consists of different levels explaining the maturity level of agile. Figure 2.3 shows ASD practices maturity levels from initial to sustained level. In the initial level, the software development process is not stable which means the ASD process is not defined in the organizations. These organizations implement their own way of developing software. Problems related to communication and software quality appear at this level.
The second level is more structured than the first level. According to Patel et al. [32], maturity at this level helps developers and customers to identify and improve

problems related to planning, RE, and on-site customer by learning from previous project successes and failures. In this level, organizations focus on the project planning, improving agile RE, and customer and stakeholders' orientation. In the project planning, project plans and schedules are created by the developers, while story cards standards are specified and written by POs.
The third level focuses on practices related to the customer satisfaction, communication improvement, software quality and enhancement of coding practices and coding standards. Organizations in this level increase the control over their practices such as coding and testing. Therefore, most of the technical problems are solved while problems related to the team are unsolved. Maturity in this level helps developers to identify and improve issues related to customer relationship, coding, testing, frequent releases and coding standards [32].
The fourth level focuses on the project management, working hours, self-organising team, and risk assessment. Maturity in this level aims for more respect for people involved in the development process [32].
The mature level is the last level, where improvements in the organizations go through quantitative feedback. Main goals for this level are defect prevention and context improvement [32].

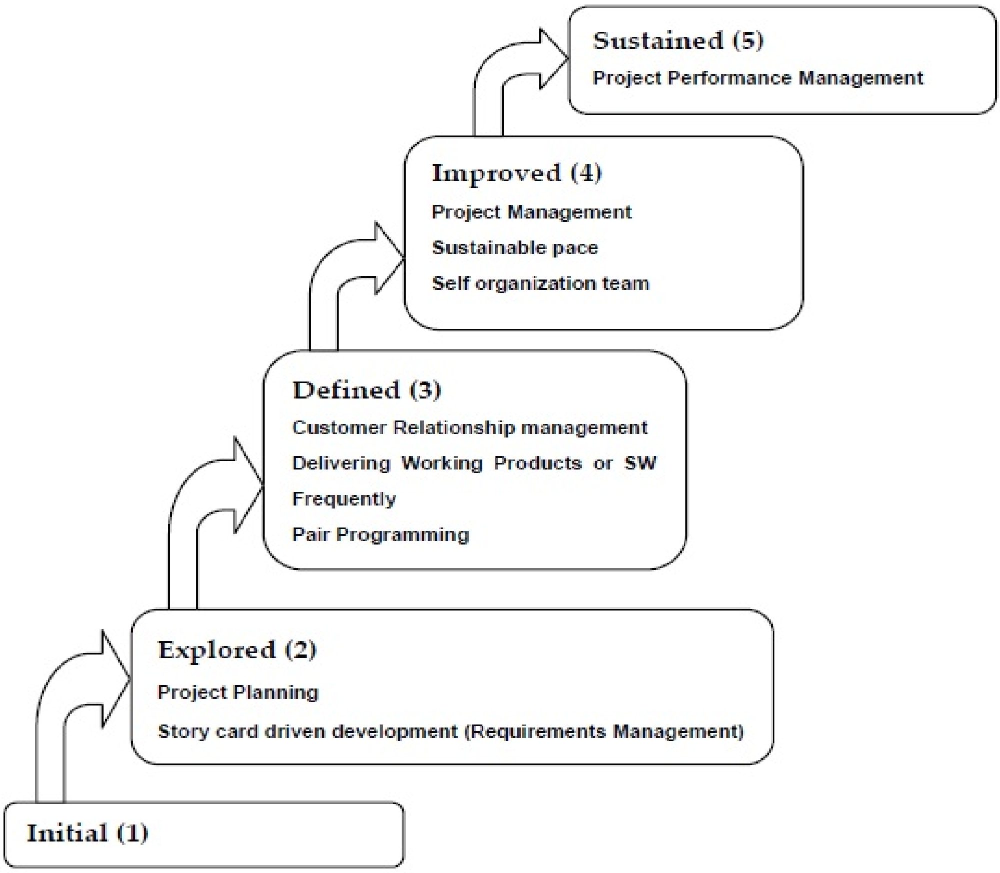
Sustained (5)
Project Performance Management
Improved (4)
Project Management
Sustainable pace
Self organization team
Defined (3)
Customer Relationship management
Delivering Working Products or SW Frequently
Pair Programming
Explored (2)
Project Planning
Story card driven development (Requirements Management)
Initial (1)

Figure 2.3: Agile Maturity Level AMM [32]

2.5.2 Capability Maturity Model

Capability Maturity Model (CMM) provides guidance on software development and maintenance. This model is designed to guide software organizations in their improvement strategies through identifying critical issues for software quality and process improvement. Five levels of software process maturity were identified for CMM [5]. Figure 2.4 shows these levels from initial till optimizing.
In the initial level, few processes were identified. The software process is characterized as ad hoc where the organizations do not provide a stable environment for development. The capability of software process, schedule and budget in this level are unpredictable [5].
In the second level, software project polices are established. The planning and tracking of software projects is stable and under control. In this model, an effective process can be practiced, documented and measured [5].
The organization's standard process for developing software is documented in the third level. This process is used to help the managers and technical staff to perform their work effectively. In this level, a wide training is implemented to ensure that both managers and staff in the organization have the required knowledge to fulfill their roles. Moreover, the capability in this level is standard [5].
At the forth level, the quality and productivity are measured. The software process in this level is well defined and consistent. It means that the capability in this level is predictable [5].
The optimizing level is the fifth level in the CMM model. The organizations can define their weaknesses and strengthen in this level. The capability in this level can be characterized as continuously improvement [5].

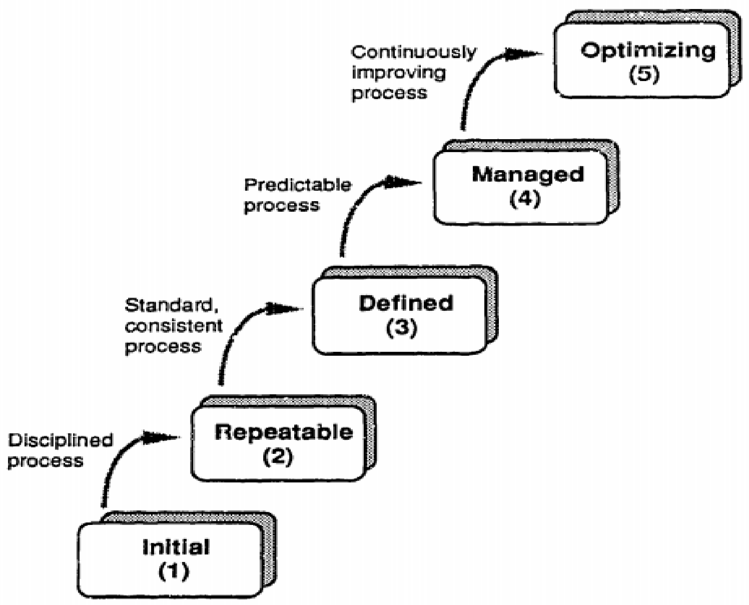

Figure 2.4: Levels of software process maturity [5]

3
Case Company

In this research, we use the IT department of AF as the case under investigation. AF is a large governmental organization in Sweden that provides services to their stakeholders and facilitates the matching between job seekers and employers by providing required skills to job seekers according to employers' needs [33]. The main responsibility for AF is to match both job seekers and employers in order to minimize the rate of unemployment. AF consists of different departments such as IT, Digital Services, Mediation departments. In 2014, the new general director of AF decided to implement the so-called 'Renewal Trip', which aims at making Sweden richer by letting people and businesses grow. Every department in the organization planned to implement the Renewal Trip from their perspective. The IT department chose to transfer to ASD.

3.1 IT Department

IT department is one of the departments in AF that is responsible for design, development, procurement, implementation and operations of IT solutions in AF. The department supports different stakeholders in AF with IT-related orders. These orders come from different sources, namely, Digital Services department and other internal departments in AF. Most of the received orders come through Digital Services department in the form of requirements. This department is responsible for all internal and external digital services for AF's stakeholders (see section 3.2). Figure 3.1 summarizes how the orders and requirements flow in AF.

Since we investigate the communication with different stakeholders, we aim to focus on both internal systems, that are, used within the case company and external systems that are, used outside of the case company, in order to get detailed insights about the studied area.

3.2 IT department Stakeholders

The IT department has different categories of stakeholders, which include External Stakeholders (ES)(job seekers and employers), customers and Internal Stakeholders (IS).

Job seekers are those who are looking for work, need to know more about the labor market, and need tips and advice on how to find their desired jobs. Employers are those who look for staff. AF provides different services to employers through publishing their vacancies announcements, arranging for recruitment meetings, and

financial support.
IT department has different customers, such as the Mediation department in AF. The customers receive instructions from the Swedish government. After that, the orders are prepared according to the received instructions and delivered either to Digital Services department for analysis with the Product Owners (POs), or delivered directly to the IT department (See Figure 3.1). In addition to that, the IT department has 500 employees composing 30 different development teams, which are considered as IS. Those stakeholders has differed roles in the IT department such as developers, testers, POs, Team leader, etc.

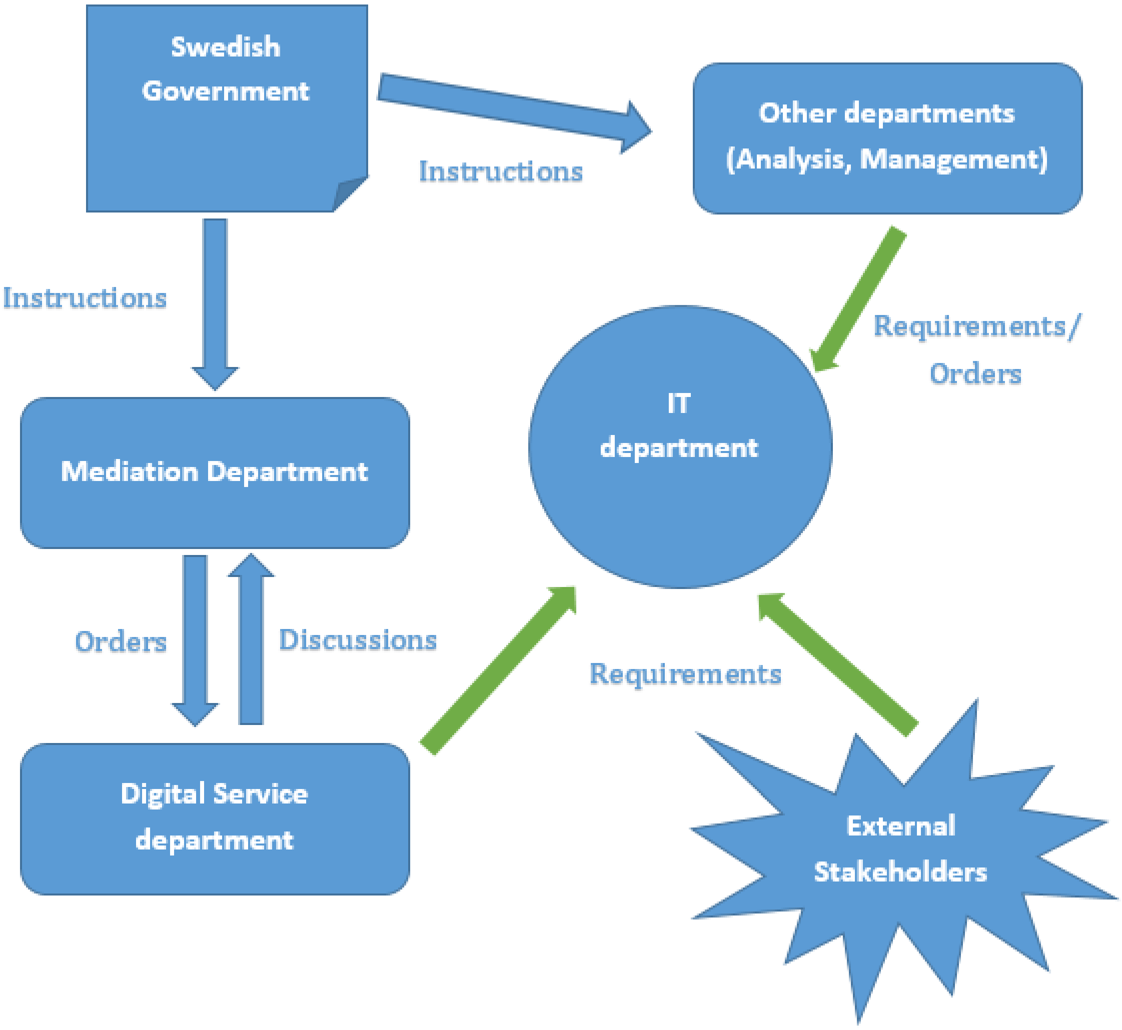

Figure 3.1: Orders or requirements flow in the IT department

3.3 Agile practices in the IT department

In 2014, the IT department started to implement ASD, with the entire department moving from traditional development to agile development. All the employees in the department moved to an open workplace. The development teams were reformed, and the employees could choose their responsibilities as well as their teams. Most of the development teams have their PO. However, many POs work in different teams. All the development teams used Scrum as agile practice. In November 2016, many development teams decided to use Kanban as a practice instead of Scrum. Two reasons are standing behind this change. The first is that the development teams cannot deliver a working product or functionality every two weeks due to top management's decision to deliver only four times/year. The second reason is that the development teams need to work on a prioritized backlog, which they found difficult. Overall, the department is still immature in implementing ASD.

The AMM and CMM we introduced in Section 2.5.1 and Section 2.5.2, consist of five levels. On the first hand, the second level in AMM model has several goals such as project planning, agile RE improvement and customer and stakeholders' orientation practices. While the third level has other goals such as customer satisfaction, communication improvement, software quality and enhancement on coding practices and coding standards. On the other hand, the organizations in the second level in the CMM model have installed basic controls for software management. Project commitments are based on the previous projects. Moreover, requirements in this level are developed and controlled. In the third level, a well defined software process identified and the management has a good insight into technical progress.

In the case company, they are learning from previous project success and failure and ensure that story cards are specified and written by POs. However, they do not qualify for the third level of AMM, because they do not focus on practices related to the customer satisfaction, communication improvement and software quality. Therefore, we can consider the case company to be on the second level of AMM and CMM. Table 3.1, describes which goals and process areas are implemented in the case company to be considered in the second level of AMM. While Table 3.2, lists some goals and key processes which are not implemented in the case company to be considered in the third level of AMM.
This means that we can consider the organization to be agile to some extent. However, there are multiple aspects that should be improved with respect to agile development. As we will discuss later on, this lack of complete adoption of agile principles and practices impacts also the challenges we found.

Practices and explored goals		
Project Planning	Agile RE improvement	Customer orientation
- POs present in all team sessions - The project is divided into the iterations	- User stories are written to solve RE - Unique story card structure is defined - Elicit customers need from POs	- POs are domain expert

Table 3.1: Goals and key process areas for AMM in the second level

Practices and explored goals not implemented in the case company		
Customer Satisfaction	Delivering working product	Pair Programming
- Effectively collaborate customer	- Made small frequent releases which will create feedback loop	- The customer pay frequent visit to the development team
- Make small frequent release	- Unique story card structure is defined	- Only one pair integrates code at a time

Table 3.2: Goals and key process areas for AMM in the third level

3.4 RE in the IT department

The IT department implements some RE activities such as elicitation and analysis of requirements. All the development teams work on the received requirements almost in the same method, which is discussing the received requirements, refining them and making them understandable for development. The discussions in several development teams take place with the POs, while in the other development teams it takes place with Requirements Engineers or Requirements Analysts. These engineers and analysts still collaborate with some development teams that work on large and complex software projects and do not have POs. Therefore, the traditional development approach is still adopted by some development teams. After the analyses, user stories are written in different tools such as Jira, ReQtest, Trello, or as Word documents. The tools are briefly discussed in Section 3.7. The IT department plans to narrow down their usage of the tools to Jira instead of other tools. The reason for using Jira is seeking for effectiveness.

The IT department is putting more focus on the user testing process. This focus is implemented through using several Interaction Designers (UXs) who are in a direct contact with the ES in order to implement user testing and obtaining feedback from the ES.

3.5 RE in traditional environment in the IT department

Before moving into adopting agile approaches, the IT department used to work in a traditional way. A requirement coordinator used to prepare and deliver requirements work packages to the development teams, which means that the received requirements were analyzed and presumed ready for implementation. The development teams had no direct communication with the coordinator, which resulted in a lack of clarity, compliance to the customer expectations, and higher misconception about the received requirements.

3.6 Benefits of moving from traditional to agile methodology

According to the IS, the IT department experienced several benefits when moving from traditional development to ASD. One of the benefits was that the development teams were involved early in the development process, which means that there was a faster progress of development. Another benefit to many development teams was that they could establish a better communication environment, both internally and externally. The communication environment was improved through conducting three different meetings, which are: a daily stand up meeting, weekly meeting and synchronization meeting. These meetings provide the opportunity for deeply discussions inside and between the development teams. An additional benefit, from RE process, is the close and tight cooperation between POs and the development teams, which lead to having well described requirements.

3.7 Tools used in the IT department

The following tools are used to handle requirements and requirements-related artifacts in the IT department.

1. Jira is a system dashboard produced by Atlassian company. Agile teams use it in order to plan, track, and release software [34]. It is used as a case tool in the IT department, in order to create user stories and issues, and prioritize the team work.
2. ReQtest is a cloud-based software development and maintenance management tool that is produced by ReQtest AB [35]. The IT department uses the tool to handle test cases.
3. Usabilla helps different companies to improve the performance of their websites, apps, and emails with direct user feedback [36]. This tool is used in the IT department to obtain feedback from their ES.

4
Research Methodology

In this chapter, we present first our research questions and hypotheses that were based on our literature review. Later, we discuss in detail our research methodology and why we used the case study approach.

4.1 Research Questions and Hypotheses

The literature review discussed in Chapter 2 shows that there are existing challenges and risks encompassed while using RE in ASD. Heikkilä et al.[10] defined the challenging aspects of agile RE as being related to the use of customer representatives, the user story requirements format, and the prioritization of requirements. Thus, in this research, we will aim to find the answers to the following questions:

RQ1: Which challenges with respect to RE-related activities appear, when implementing agile practices in software engineering?
RQ2: How do these challenges relate to communication with stakeholders?

Consequently, we have also derived three general hypotheses for our research. These hypotheses will be used as guidance for designing the interview questionnaire. We mentioned in Chapter 2 that Heikkilä et al. [10] identified different problems related to RE such as the use of customer representatives, the user story requirements format, and the prioritization of requirements. From this, we can derive the first hypothesis.
H1: Implementing RE in agile environment has a negative effect on requirement prioritization

Bjarnason, Wnuk, and Regnell [14], suggested that the use of some agile practices, such as customer representative, planning game, and motivating teams in RE, results in communication gaps and overscoping. Accordingly, we derived the second hypothesis.
H2: Large companies that use ASD suffer from communication gaps and overscoping.

Sillitti and Succi mentioned in their book that the different perceptions among stakeholders, in agile methods, results in a complex interaction between internal and external stakeholders [19]. Thus, the third hypothesis was derived.
H3: Agile environment has a negative influence on the relation between internal and external stakeholders.

The aim of this research is to study the combination of ASD and some RE activities in large organizations and investigate the communication between different stakeholders in agile RE. To answer our research questions, semi-structured interviews, telephone meetings, and email contact were prepared to collect our data from our stakeholders.

4.2 Case Study

Empirical studies have recently gained popularity in Software Engineering research [37]. There are two different approaches for running empirical studies, quantitative and qualitative. Since this research aims at investigating the challenges encountered in implementing some RE activities in ASD, the qualitative approach was selected as an appropriate one to facilitate our investigation. Qualitative methods have been developed to handle the complexity of issues that involve human behavior such as motivation, communication, and understanding [37]. Data collected from qualitative methods are represented in the form of words and pictures. The main advantage of using qualitative methods is to enable the researcher to delve deeply into the case rather than abstract it. Fuzzy results and difficulties in the summation are the drawbacks of using these methods [37]. Runeson et al. [38] discuss different research methodologies research methodologies and their use in software engineering; namely Survey, Action Research, and Case Studies. Mentioned below are definitions of these research methodologies.

Survey Research: is "a collection of data representative of a population. The gathered data is used to generalize findings from a sample population within the limits of random error" [39].

Action Research: is "attempting to solve a real-world problem while simultaneously studying the experience of solving the problem" [38].

Case study: is defined by Runeson et al. [38] as a research strategy in a particular area to increase knowledge and understanding to improve the process and results of studied area. Although, case studies are assumed as difficult methods in empirical studies, they offer deep understanding of how a certain phenomenon occurs [40]. In software engineering, a case study is a suitable method to answer different research questions in a multidisciplinary area. The main weakness of case studies is the open interpretation and the research bias of collected data [40]. Runeson et al. [38] define five steps, which are shortly outlined in section 4.3. The steps are design, preparation for data collection, collecting evidence, analysis, and reporting.

As we mentioned previously, the aim of this research is to investigate the challenges that arise when using RE-related activities and agile together and how those challenges relate to communication between stakeholders. To reach this aim, detailed insights are important in order to understand how and why these challenges exist and how they could be addressed. Therefore, we chose the case study approach since it is appropriate for this research to investigate these challenges while the results can offer new insights to researchers in the research field.

4.3 Research Design

According to Runeson et al. [38], the research design for a case study is a work plan which can be documented as a protocol. Different elements should be considered in the design of any case study. Some of these elements will be discussed in detail in the following sub-sections, while others were already discussed in previous sections such as theoretical framework, research questions, and hypotheses.

4.3.1 Objective of the Study

The objective of this research is to investigate the challenges that arise when using some RE activities and agile together and how those challenges relate to communication between stakeholders.

4.3.2 The Case

The IT department in AF lacks effective communication between internal and external stakeholders. Thus, in this research, the IT department of AF was studied to investigate these issues.

4.3.3 Methods of Data Collection

The principal decisions for choosing the methods in the data collection are taken at the design time of the case study. Using multiple data sources, creating a case study database, and validating data are the principles for data collection [38]. According to Lethbridge et al. [41], there are three categories of methods for data collection, which are direct, indirect, and independent.

Direct Methods. In this method, researcher and interviewees make direct contact with each other to achieve the data collection objectives. For our research, we interviewed twelve IS with different roles, such as developers, product owners, and testers from IT department in AF. We also interviewed two external stakeholders and two customers.

Indirect Methods. In this method, without direct contact between researcher and interviewees, raw data is collected directly from their resources. For this study, we have collected some data from the tools that are in used by the IT department such as Jira and ReQtest (see section 3.7), which helped us forming an overview of how requirements and user stories are handled in these tools. We got access to Usabilla to understand how the IT department gets feedback from their external stakeholders.

Independent Methods. In this method, the researcher analyses available work artifacts. For this study, we used this method through accessing some documentation to see in which format the requirements are received. Our conclusion was that most of the requirements come in the form of written documents from customers.

4.4 Data Collection

In our research, we used the three methods of data collection that were described previously in sub-section 4.3.3. In this section, we present the research setting and the research participants as well as the interviews and the content of the interview questions.

4.4.1 Research Setting

The research setting for our research is AF. We chose to study the IT department that is based in the headquarter of AF in Stockholm.

It is important to prepare our interview questions as good as possible. To do so, we carried out a pilot test on our interview questions in order to validate the reliability of the questions. The interview process was conducted on-site during the first week of the data collection phase. We interviewed six interviewees from the IT department who are not included in our later data collection. After doing this pilot, we updated some interview questions, added new ones, and discarded others.

4.4.2 Interviews

Interviews are one technique for collecting qualitative data. They are used to collect historical data, opinions or impressions about something and to identify the terminology used in a particular setting [19]. There are several types of interviews: structured, semi-structured, and unstructured interviews. Structured interviews are in a form that the interviewer asks a set of predefined questions to the interviewee. In unstructured interviews, the interviewer derives the questions from the response of the interviewee in order to elicit as much information as possible about some issue. Since unstructured interviews are costly, semi-structured interviews are more common in many studies [42]. This type of interviews includes both specific and open-ended questions. We also arranged short telephone interviews with the external stakeholders.

Before we started our interviews, two criteria were set. The first one was to record the interviews in order to use these records in our analysis. The second was that all interviewees should accept to be contacted after the interviews in case of unclear answers.

4.4.3 Interview Questions Content

This section discusses the four different sets of interview questions that were asked to the stakeholders (see Appendix A, B, C) for the full interview questions. The questions for the IS were divided into five different scoped areas. First, questions one to four covered general background information about the interviewee such as role, experience, and knowledge of agile practices. Second, question five focused on RE to understand how RE is implemented in the IT department and which challenges and benefits exist in the RE process. Third, question six to seven focused on ASD to understand which benefits and challenges emerge from implementing ASD in the IT department. Fourth, question eight to thirteen focused on identifying the communication channels between the stakeholders, challenges, and to suggest ways to overcome these challenges. Finally, the last question was the ideal picture of the IT department, from the interviewees perspective.

The managers were asked two questions that are related to the challenges of implementing ASD in the IT department and about the challenges related to the communication between the stakeholders, from their viewpoints.

The customers were asked six questions; each aims at understanding how he or she communicates with the IT department, how he or she writes the requirements and what challenges relate to the communication with the IT department and POs.

Two short questions were asked to two external stakeholders, a job seeker and an employer, over the phone. The questions aimed to understand how they communicate with IT department and which feedback they send to IT department.

4.4.4 Research Participants

Our sample size comprised of ten IS from the IT department, two customers, and two external stakeholders. Two managers from the IT department were interviewed as well for more information about challenges from implementing ASD and how those challenges affect the communication with different stakeholders. The IS were randomly selected from different development teams by a gatekeeper, which means that the author had no personal relationship with any of the interviewees. The selected IS had different roles in the department such as developer, tester, PO, and UX. Moreover, the interviewees' experience varies in between long and short experience. In this research, individuals who have had worked for four years or longer in the IT department were regarded as long experienced as they experienced both traditional and agile development environments. On the other hand, individuals who have had worked for less than four years were regarded as short experienced. The customers were also selected from the Mediation Department. Moreover, two external stakeholders have been chosen randomly from one of local AF offices in the Sweden to get insight into their experience in the communication with the IT department. Table 4.1 shows our data sources in this research related to experience in the IT department, where we grouped them as long experienced "L.Exp" and short experienced "S.Exp".

4.5 Data Analysis

One characteristic of research strategies is triangulation, which involves taking multiple perspectives to increase the validity of the findings [38]. In our research, we designed our research questions and derived different hypotheses after reviewing the literature. We did semi-structured interviews with different stakeholders in order to

Position	Experience in Years	Label	Way of contact
PO	9		
Agile Coach	6		
Tester	6	L.Exp	Interview and Email contact
UX	5		
Tester and test leader	4		
Developer	2		
Graphic Designer	1.5		
Requirements Engineer	1	S.Exp	Interview and Email contact
Requirements Engineer	1		
UX	1		
2 Managers			Interview
2 Customers			Interview and Telephone contact
2 External Stakeholders			Telephone contact

Table 4.1: Data sources related to experience in the IT department

investigate the challenges of using RE in agile. We derived some themes from interviews results which would relate to our hypotheses as shown. We have used data triangulation in our research, through using several types of data sources to limit the effects of one interpretation of one single data source. In section 4.3.3, we describe our methods of data collection and from which sources the data was collected.

For our research, we used thematic analysis to analyze the collected data. Thematic analysis is one of the fundamental methods of qualitative analysis with the aim of identifying, analyzing, and reporting patterns (themes) within data [43]. It consists of five phases which are described in the following subsections.

4.5.1 Data Familiarization

The goal of this phase is to get familiar with data collected. The interviews have been recorded, and some notes were taken as well. We did a summary of all the interviews after listening to all recordings, and the content was reviewed to get a deep understanding and to conduct a thematic analysis.

4.5.2 Initial Code Generation

In this phase, initial codes were created. Each interview transcript has been printed out and reviewed again in order to link the data collected to our research questions. We identified related parts, and they gave an appropriate label such as "Challenges of RE in the agile environment", "Challenges of using agile", "Problems of communication", "Communications channels". "Detailed requirement”, and "Business

Requirement".

4.5.3 Searching for Themes

After initial codes were generated in the previous phase, we went through the generated labels and identified common themes, which are: "Clarity of requirement", "Responsibility", "Prioritization", "Flexibility", "Resistance", "Undetailed Requirement", "Lack of Knowledge", "Release Frequency", "Informal communication", "Lack of Physical Communication", "Testing" and "Use of Customer's Representatives".

4.5.4 Reviewing Themes

In this phase, we reviewed the identified themes in the previous phase to ascertain if the themes needed modification or if there was a need to include new themes. At this point, there is no need to add new themes or update.

4.5.5 Defining and Naming Themes

In this phase, we identified the relation between the identified themes and our research questions. According to that we categorized our themes into two different categories: "Challenges of agile RE" and "Communication Challenges"

4.6 Threats to Validity

In qualitative research, there are five related threats to validity which are descriptive, interpretive, theoretical, generalizability and reactive validity [44]. In the following, we describe these threats and countermeasures we took in our research.

Descriptive Validity. For our study, there is some bias because gestures, pauses or stress. All the interviews with all stakeholders were summarized and transcribed after listening to the records in order to use the transcripts in the analysis. These transcripts might be descriptively invalid because we might do not capture some behavioral aspects which threatens the descriptive validity. To ensure this validity, the recordings are used.
The interviews were conducted in English, however, we got some answers in Swedish. We aimed to keep the meaning close to the original. However, some inaccuracy could affect the meaning, because the researcher is not native Swedish speaker.

Interpretative Validity. In our study, there are different stakeholders. Before we started our interviews with the stakeholders, we ensured that there was a shared understanding of agile and communication between IS and customers as well as communication aspects between external stakeholders. To increase validity, the results are discussed with the stakeholders in order to see if we interpreted something wrong.
Since the researcher works in AF as an employment officer, it can be a threat to the

research. The researcher has a view on the IT department and has a background about the studied problem. Moreover, the IS interviewees could hide some information that could have a negative effect on the results because they know that the researcher is working in AF.
In order to increase the validity, the researcher tried to keep the relation with IS much formal and focuses on the related tasks.

Theoretical Validity. In our interviews with the stakeholders, we asked questions in a slightly different ways in order to make the questions more understandable to the subjects, e.g., by using examples. This construct in the interpretation is a potential threat to theoretical validity. We have also built our preconceptions according to information from the IT department that they lack good communication with stakeholders. Thus this could threaten the validity of the research by interpreting the received data from this perspective.
Another risk resides in the subjectivity of categorizing the challenges, such that other researchers might come up with different categorizations.

Generalizability. In our research, we study a phenomenon in an organization. Our finding may apply to similar organizations and can not be generalized to all organizations. However, we used triangulation using different data sources such as direct, indirect, and independent in order to strengthen the validity.
Regarding what we mentioned in Section 3.3, the organization is on the second level of the AMM and the CMM. This low maturity could limit our conclusions even if the participants have a good perception on ASD.

Reactivity Validity. The influence of the researcher on the interviewees and the research setting is critical and can not be avoided. We implemented pilot interviews in order to test and correct any biased questions.

5 Research Findings

In this chapter, we present our findings for the questions mentioned in the previous chapter. We cover our findings for RQ1 in section 5.1 and RQ2 in section 5.2.

5.1 Challenges of RE in Agile environment

In this section, we present our findings for the **RQ1** (which challenges exist with respect to implement RE-related activities in ASD). These challenges are illustrated in following sections. The most frequent challenge is Unclarity of Requirements, which was highlighted by seven of ten interviewees.

5.1.1 Clarity of Requirements

In the IT Department, the challenges related to clarity of requirements play a critical role, because they affect the development process. Seven out of ten IS interviewees (four L.Exp and three S.Exp) receive unclear requirements from customers or their representatives. According to the interviewees, the lack of clarity of requirements comes because of different factors, namely Lack of the Domain Knowledge, Undetailed Requirements, and Lack of Experience of Writing Requirements.

Lack of Domain Knowledge. As mentioned previously, the requirements are obtained directly from customers or from their representatives. Three IS interviewees with S.Exp stated that requirements are received in a business form and are often written in a complicated way. In many cases these requirements are in need for more time and effort to be analyzed and converted into technical requirements to be understandable and usable by team members. For example, one interviewee said:

> *"The requirements are written in ReQtest by some administrator. When I get notice about the new requirements from the administrator, I read them, and in most cases, they are complicated"*

Another interviewee stated that the received requirements from the customers are business requirements. The orders come verbally from the customers where in several cases, the interviewee is forced to make different contacts with Requirements Engineers from different teams to get more knowledge about the customer's need.

The two customers we interviewed confirmed that they send the requirements in the form of business requirements. The second customer stated that, although he sends the requirements in a technical form, the rest of the customers do not have the knowledge about technical requirements. In other words, he emphasized on the fact that most the customers send their requirements in a business form. In summary, requirements authors currently formulate requirements in a way that is not understood by the development teams.

Undetailed Requirements. Five IS interviewees (three L.Exp and two S.Exp) who are two testers, two UX, and one developer, stated that the requirements received from the POs are undetailed and in need for extra discussion. The discussion provides more information about the received requirements before starting the development process. The developer pointed out that they have a Requirements Analyst who works on the received requirements from the POs to make them understandable and more detailed to the development team. However, he said that the requirements are still complex because of the integrations between different systems which make the requirements complex and hard to break them down.

Lack of experience of writing Requirements. Three IS interviewees, two testers, and one UX mentioned that either the received requirements are not written in a good way to be understandable, or the requirements are not ready to be used. This means that the customer needs are unclear. They added that the POs lack experience in writing the requirements. One tester stated that the received requirements are very big. The big size of the requirements needs much time to be implemented. This leads to an inability of delivering a working product by the end of the sprint which is a direct conflict to the agile practice of delivering a working increment after each sprint. Other interviewees pointed out that they receive the requirements verbally because the customers do not have experience in writing technical requirements.

5.1.2 The Use of Customer Representatives POs

IT Department receives the requirements from different sources such as POs and customers, see Figure 3.1. Seven IS interviewees (four with L.Exp and three with S.Exp) stated that they receive the requirements from POs, who deliver the requirements to be analyzed and discussed by the Requirements Analysts or Requirements Engineers. The interviewees pointed out that they do not have a direct contact with the customers and therefore also no feedback from the customers.

Several POs work on different products in the same team. They have not a prioritized product backlog which leads to a big conflict in the communication between the development teams or inside the team. One interviewee who is a PO mentioned that they do not have the opportunity to create enough physical communication with the customers in order to get feedback in the earlier stages of the development. The PO suggested that the whole development teams should meet the customers to get the feedback. The customers on the other side pointed out beside their lack of knowledge of the technical requirements or agile methodology, there is a lack of communication between them and the development teams where in many cases the customers did not send any feedback for a delivered product. In addition to that, the customers stated that because of the overlapping between the delivered order for development and preparing for the next order, there are limited chances to send feedback.

Three IS interviewees (two with S.Exp and one with L.Exp) stated that the orders come from the customers to the development teams where Requirements Engineers receive them and make them clear before the development process. This means that having the Requirements Engineer role from traditional development helped to overcome the difficulties introduced otherwise.

5.1.3 Too Much Flexibility

The IT Department suffers from too much flexibility which affects the communication inside the teams or between the teams. They transferred to ASD since 2014, but they still suffer from undecided work practices. Five IS interviewees (three with S.Exp and two with L.Exp) stated that they suffer from this flexibility because everyone can decide the work methodology without any guidelines. One interviewee said:

"We skip the analysis meeting recently because we are a small team and we are close together"

While another interviewee said:

"We do not have any working way to manage our work. Everyone decide which strategy to be used"

This flexibility affects the process by developing uncompleted products to the customers, providing that everyone decide how to work and when to develop. One interviewee said:

"Everyone takes a small part or feature and work on it, and we have totally seven features which are not finished yet (everyone works on his own) so, there are no completed features"

The amount of team members is another type of this flexibility challenge. In IT department, there are between 26 to 30 development teams. Some teams consist of eleven members where others consist of four members. This variation of the team building affects the coordination of the teams. One interviewee from a team of eleven members stated that the teaching of agile takes a long time to be understandable because of their team size, where another interviewee from the team with four members emphasized that they suffer from not updating the requirements in case of actual updated on the requirements. This interviewee refers this problem to the small team size. Hence, the differing team sizes result in very different problems that can't be addressed centrally.

5.1.4 Understanding of Agile Still Weak

As we mentioned previously in section 5.1.3, the IT Department started use agile since 2014. However, some employees in the department, the highest level in the organization and many customers still have a lack of awareness of agile. Though the problem of the awareness of the agile exists, the IT department has only one time published and distributed a brochure about agile to spread the knowledge among all the employees in the department. All the interviewees and the management level stated that the agile practices in the IT department are not implemented consequently enough. One IS interviewee with S.Exp stated that they started ASD in her team three months ago. Another one with L.Exp stated that they do not have a Scrum Master. Where the third interviewee with L.Exp stated that they do not do enough tests through the development process. He said that:

"We test by the end of the sprint. We still don't have an agile mentality"

Although the management level in AF is not agile, they decided when and how products shall be delivered. Two of the managers in the IT department and four IS interviewees (two with S.Exp and two with L.Exp) emphasized that the top level in the organization is not agile. One interviewee said:

"We still have certain processes within the organization that aren't really agile It means that sometimes you have to wait for a decision from the highest level within the organization to do something"

An additional challenge is what could be called a low release frequency. This challenge can be referred to two main reasons. The first reason is a previous decision from the management level in AF, which restricts the delivery per year to only four times. Thus, the IT department can deliver only something working to the customers every three months. Six IS interviewees have confirmed this fact. The second reason is that many customers lack agile knowledge. Therefore they do not know about the frequent release, or providing feedback frequently to the IT department. Related to that, many teams in the IT department moved toward Kanban methodology instead of Scrum because as they said:

"We are not limited with two weeks' sprints"

Furthermore, there is still resistance toward implementing agile inside the department. Five IS interviewees (four with S.Exp and one with L.Exp) pointed out that they still have Waterfall mentality in their teams or they still working Waterfall and did not move to agile. The Waterfall mentality is represented in the existence of Requirements Engineers or Requirements Analysts. And it also appears clearly through implementing test cases by the end of the sprint which means, according to the interviewees, that many bugs exist while they do not have time to do changes and updates.

In summary, the IT department has introduced agile in order to overcome challenges they faced from implementing traditional development methodology. However, they have not implemented the agile work methodology properly.

5.1.5 Prioritization of Requirements

The last challenge related to RE in ASD is a lack of requirements prioritization. Four IS interviewees stated that the received requirements from the customers or their representatives are not prioritized. In addition to the four interviewees, two customers confirmed that they send the requirements without prioritization. These requirements come in the form of a big order from different customers, which make the development teams confused about the priority of different requirements. The development teams take responsibility for prioritization and from their side try to prioritize the requirements which means more time, effort and uncertainty about the prioritization process. Hence, the development teams should interpret the customers' need and take responsibility for prioritization; they do not always present the actual customers' need, which leads to delays in the development process. Although prioritization is not necessarily a compulsory principle in agile, the availability of the customer is essential for planning which set of requirements need to be implemented.

5.2 Communication with Stakeholders in agile environment

In this section, we present our findings for the **RQ2** and how the challenges, which identified in the (section 5.1), are related to the communication with the stakeholders.

5.2.1 Communication with Customers

IT department lacks the physical communication with the customers. Seven IS interviewees stated that they do not have any physical communication with the customers. Instead, they communicate with the POs. Moreover, the results show also that there is a lack of the communication between the customers and the development teams because many customers lack the understanding of technical issues. In other words, the customers avoid making communication with the IT department, because firstly, they do not have the time to follow up their orders and secondly, they lack the understanding of technical issues. In both situations, there is reluctance because the customers think that they cannot help through their feedback.

5.2.2 Communication with External Stakeholders

The communication with the ES exists in a few occasions where the IT department needs to conduct user tests with them. Five IS interviewees with L.Exp stated that there is a lack of the physical communication with the ES. However, the interviewees recited that the IT department often gets a negative feedback from the ES due to the weak relationship between them and the AF. The IT department started to use UXs to contact the job seekers and employers, to get more feedback on a delivered issue. For triangulation of data, we conducted three phone interviews with two ES. Both acknowledged that there is a lack of physical communication with AF, especially with the IT department. They suggested having periodic meetings with the IT department to discuss the latest updates and the external stakeholder's needs.

5.2.3 Communication with Internal Stakeholders

Communication inside and between the development teams faces challenges. All the interviewees stated that they have a good relation between each other and between their teams. Nine of them stated that they have different meetings such as daily stand-up meeting, weekly meeting with POs to follow up the development process and synchronization meeting with other teams. Regardless the good relations, the large number of teams makes it hard to create efficient and effective communications between the teams. This low communication efficiency was pointed out by the interviewees; they highlighted that it is hard to know which team is responsible for a specific product. The interviewees stated that the existing number of teams in the IT department (30 teams) makes it hard to get information about some product. As mentioned, they face difficulties to know which team is responsible for a product. This challenge affects the development process and slows it down.

5.2.4 User Testing

User testing is another issue related to communication with external stakeholder. Four IS interviewees stated that user testing is done by the UXs who get contact with the ES to test some delivered issues. However, the interviewees stated that this type of communication is not enough and the ES should be more involved from the start of the development process in order to get earlier feedback.

6
Discussion

In this Chapter, we discuss the findings of our research, which are presented in Chapter 5, and link these findings to the related literature presented in Chapter 2. This chapter consists of two sections. In the first section, we discuss the challenges of using some RE activities in an agile environment (see Table 6.2), while in the second, we discuss how the discovered challenges from implementing RE in the agile environment affect the communication with stakeholders (see Table 6.3). Table 6.1, classifies which challenges should be seen in a purely agile context and which ones are more generic for RE.

Agile Context	**Generic RE**
- Undetailed Requirements - The Use of Customer Representatives POs - Too Much Flexibility - Prioritization of Requirements - Understanding of Agile Still Weak - User Testing	- Lack of Domain Knowledge - Lack of experience of writing Requirements

Table 6.1: Challenges in agile context vs generic for RE overall

6.1 Challenges of RE In Agile environment

If we take into consideration that using agile in the IT department started in 2014, as well as the size of the department, it can be concluded that using agile in the IT department is immature. As we mentioned in section 2.5.1, agile at the department is on the second level of the AMM [32]. They implement some agile practices such as using POs, Scrum meetings and Sprint meetings. However, they still lack the knowledge about all agile practices. Two reasons are standing behind this problem. The first one is that the employees in the department have not gotten enough training on how to implement agile successfully. The second reason is that the management level in the organization lacks the domain knowledge about agile which affects the successful implementation of agile.

In addition to maturity fact, the IT department implements both traditional and agile RE activities. On the first hand, some traditional RE activities such as elicitation and analysis of requirements are used in the IT department. They still use some traditional RE aspects such as the roles of Requirements Engineers or Requirements

Analysts to overcome some agile challenges such as prioritization of requirements or unclear received requirements. On the other hand, the department uses agile RE-related activities such as user stories and acceptance testing. According to Paetsch et al. [8], implementing agile RE-related activities needs highly skilled people because of the vague definition of development processes, while traditional approaches provide more guidance to do the right thing.

The IT department faces several challenges when implementing RE-related activities in the agile environment. Two reasons could stand behind these challenges. The first reason is the low level of maturity of agile in the department. The second reason is that the employees in the department are not skilled in implementing these activities.

Table 6.1, classifies which challenges are in agile context and which are generic for RE. One of these challenges, which affects the IT department, is clarity of requirements. In Chapter 5, three different factors that were related to clarity of requirements were identified. The first one is lack of the domain knowledge, which was mentioned by short experienced interviewees, ones who had less than two years of experience in the IT department. Although this factor was confirmed by the customers themselves, while long experience employees did not even mention it. This challenge is a general RE problem and not related to agile because the lack of domain knowledge can affect any development process even if the implemented process is agile or Water-fall model. It could be concluded that the long experienced IS have either improved and developed a good relation and better communication mechanisms with the customers or they have gained more domain knowledge over time. This enabled them to understand the received requirements regardless of their form. Our finding matches the results drawn by Maiden et al., [45], which state that domain knowledge is needed to avoid requirements misunderstanding, which leads to poorly produced systems. Moreover, Sillitti & Succi [15] argue that poorly produced systems are caused by the absence of feedback about the delivered products. By comparing our finding with Sillitti & Succi's finding, it is possible to argue that lack of domain knowledge is the reason standing behind Sillitti & Succi's finding. In other words, developers who do not understand the requirements cannot provide valuable feedback. An improvement to the third level of CMM could also tackle this challenge, by implementing trainings that allow employees to fulfill there roles, in this case to train their domain knowledge."

The second factor is Undetailed received requirements. This challenge is related to agile context, where implementing agile means less documentation, therefore, the received requirements from the POs are undetailed.

Some teams in the IT department use Requirements Analysts to make the received requirements understandable through implementing a classical way of development, which means more requirements breakdown and more detailed documentation. Using Requirements Analysts means also that a traditional development concept is utilized in the agile environment. In other words, the IT department overcomes some agile challenges through implementing the traditional development methodology.

The last factor, lack of experience of writing requirements was mentioned by long experienced interviewees. This factor suggests that lack of experience in writing requirements from customer's side is causing lack of clarity in the received requirements. Hence, regardless of the receivers' domain knowledge, a substantial effort is needed to understand the requirements. It can be concluded that this challenge is related to RE. This might also be related to the fact that the general maturity of the organization's software development process is rather low, as discussed in section 3.3.

The second challenge of using RE in the agile environment is the involvement of POs. This challenge is related to Agile context because using POs is agile concept. In agile environment, there is a direct contact with the customers which is missing in the IT department. This gap in the communication with the customers could be happened in the department because they do not managed to implemented agile completely. The related work emphasizes that having on-site customers or customers representatives is a problematic area in agile RE [10],[9]. Moreover, involving a real customer in agile development is often not performed in a good way [46]. Using either POs or real customer is a complicated dilemma, since a perfect communication would not be achieved either due to the indirect communication through POs or due to the incomplete involvement of the real customer. The IT department confirmed that using POs instead of real customers creates a gap in the physical communication between the development teams and the customers. As a result, the delivered products do not, most of the times, match the customer's needs. Moreover, since different POs work on various products on the same team, significant conflicts in the communication between members of the same team or other teams were generated.

One interesting finding in this research is too much flexibility which is related to agile context. The IT department planed to implement agile in an effective way, but they instead have not planned enough to implement agile completely and correctly. This challenge comes because of implementing ASD and would not appear in traditional software development. The IT department implements ASD in order to make a change, but they in the department lack guidelines and therefore decide how to work by themselves without following established agile practices. Marchenko and Abrahamsson [27] identified some challenges related to Scrum practices such as lack of management over individualism. The lack of management increases the individualism and lead to too much flexibility. Therefore, the effectiveness of the development process is affected, where every member of the team decides how to work, and when to develop a functionality.

Weakness of understanding of agile has a negative impact on the development process and causes poorly delivered products. The weakness of understanding is evident among the customers, IS, and the top management in the AF. The customers lack the knowledge about agile practices such as face to face communication and continuous feedback, while the IS have some understandings but still not enough. Ultimately, people from the top management in the AF have no knowledge about agile practices resulting in limiting the delivery of software products to four times per year. The weakness of understanding in agile resulted in creating a challenge of low release frequency. This lack of understanding in agile practices across the different stakeholders seems to have significantly affected the development process and the communication.

The final challenge of using RE in agile we discovered is prioritization of requirements. This challenge is related to agile context where in many agile processes, prioritization is done with the customer which means that it does not have to be that all requirements are prioritized, but instead the customer has to be available. According to Ramesh, Cao, and Baskerville [4], requirements prioritization supports the analysis by identifying the valuable requirements. The POs do the prioritization through collaboration with the development teams [4]. The IT department receives unprioritized requirements in form of business requirements. Some of the development teams try to prioritize the received orders which means more wasted time, effort and uncertainty about the prioritization process.

Our findings related to RE in agile environment encourage the devotion of more time of training to the IS on agile practices. This could increase the effectiveness of the development process. Moreover, the customers and high level management in AF should be introduced to the agile practices that are being implemented in the IT department and ensuring that they are committed to agile.

Agile RE challenges	**Related literature**
Clarity of Requirements	[45], [15]
The Use of Customer Representatives POs	[10], [9], [46]
Too Much Flexibility	[27]
Understanding of Agile Still Weak	
Prioritization of Requirements	[4]

Table 6.2: The research finding: Agile RE challenges and the related literature

6.2 Communication with Stakeholders

Our results presented in section 5.2 show that the IT department suffers from lack of communication with different stakeholders. Since agile practices motivates good collaboration between different stakeholders, it is therefore seen as the best solution for overcoming communication problems manifested within software development teams [17]. According to Sillitti and Alavi [6],[20], lack of communication between the stakeholders leads to failures in the development process and can have serious consequences in meeting customers' expectations. This is aligned with the results drawn in our research since the IT teams were experiencing failures in the development process.

The following paragraphs give insight about the communication between different stakeholders, namely customers, external and IS.
Firstly, regarding communication with the customers, there is an absence of a physical communication between them and the IT department. The customers do not have enough time to follow up their orders, and they lack the understanding of

technical issues. These two factors create a gap in the communication between the customers and the IT department.
According to Table 2.1, customer availability is one of agile RE challenges. The proposed solution for this challenge is to have a Proxy customer role from customer's side, who is responsible for communication with development teams [13]. Based on the inputs of this study, it can be proposed to create a new role in every development team through assigning an ambassador to collect feedback directly from the customers. The ambassador can be one of the team members or a newly assigned member. The ambassadors should have sufficient domain knowledge, as they have to be able to understand the business view. This solution could enhance the communication between the customers and the IT department and minimize the dependencies on the POs. As a result, the customers' expectations are more likely to be met. One risk from this solution is the absence of the ambassador because of sickness or leaving the company. This problem can be solved by having a backup ambassador who can take the responsibility.

Secondly, the communication with the ES is weak and in need for more development. According to Trust Barometer and AF report [47],[48] the relation between AF and job seekers suffers from a low level of trust, because of the shortage of provided services. While the AF report was published before the agile transition, several interviewees still mentioned that the relation between AF and ES is in need for development. This negativity in the relation between AF and ES affects the development process in the IT department, which can be measured in the negative feedback or lack of this feedback. The IT department started improving the relation with the ES through using UXs to involve ES in the development process, mainly for user testing. In addition to that, the department uses Usabilla (see section 3.7) to communicate with ES to get the feedback on delivered products. However, several interviewees mentioned that these actions did not lead to improved communication with ES. The interviewees from the IT department propose involving ES early in the development process. Moreover, the ES recommend having periodic meetings with the IT department to participate in the development process.

Lastly, the improvement in communication between the IS is a direct result of carrying out an assortment of meetings, such as stand-up and synchronization, which increases the formal communication within the teams. Moreover, the IT department has an open office space. Therefore the obstacles were removed to have direct communication with the team members. However, communication between the development teams faces some challenges because of a large number of development teams and the structure of these teams. The large scale of the IT department hinders the communication among IS as well as among other stakeholders. The interviewees from the IT department emphasize the importance of developing a knowledge database which contains information about all development teams and that guide all development teams to find information quickly about the development process as well as the development teams in order to simplify communication between the teams. Interestingly, this would mean introducing more documentation, in contrast to the ideas of agile development. Another proposed solution for communication problems inside the teams is to have a small team composition. This is in line with the findings of Sillitti & Succi's [6] who write "small teams are more agile than large teams."

The IT department at AF, as a part of a Scandinavian organization, has a strong focus on formal communication [49]. Still, it can be clearly noted that the IT department has improved and relied on its informal communication thanks to the implementation of agile. Hence, organizations implementing agile practices seem to benefit from increased informal communication even if they have a culture of strong formal communication.

Communication with Stakeholders	**Related literature**
Communication with Customers	[13], [2], [1], [17], [16], [18]
Communication with EX	[47], [48], [2], [1], [17], [18]
Communication with IS	[6], [49], [2], [1], [17], [18]

Table 6.3: The research finding: Communication with Stakeholders and the related literature

7
Conclusion

According to related work, there are several challenges associated with using RE in an agile environment. Those challenges could have an adverse effect on the communication with stakeholders. However, there is a lack of publications that study how communication with stakeholders and the use of RE in agile development are connected.

In this research, we presented our results from a case study with the IT department in AF, the Swedish public employment agency. The challenges of using RE-related activities in ASD have been studied, and the effect of these challenges on the communication with different stakeholders has been investigated.

It can be concluded that the studied organization is not agile enough and they still mature. Using agile in the studied organization is in the second level of the AMM maturity scale. They are learning from previous projects success and failure. However, they do not focus on practices related to the customer satisfaction, communication improvement and software quality.

The research shows that there are several challenges from using RE in an agile environment, which affects the communication with stakeholders. Those challenges are not related to the agile context and are generic for RE (see Table 6.1). These challenges were examined in this research which shows that communication with different stakeholders is, negatively, influenced by these challenges.

One key challenge that relates to agile context is a lack of direct contact between customers and development teams. Additionally, while communication with ES is still weak, communication with IS needs more improvement. Existing communication links between development teams and different stakeholders are weak and affect the quality of delivered products.

Another challenge identified in this research is a lack of awareness and adoption of agile methods. Our research shows that agile practices need to be implemented by different parties which are high level of management; development teams; and customers. First, the high level management should have a general awareness of agile practices in order to give enough space and flexibility for development teams to implement agile in best practice, and to avoid the limitation of product delivery. Second, development teams, which are mainly responsible for implementing agile practices, need more training and coaching to be able to conduct agile in most efficient and effective form. Development teams should be able to lead the agile practice within the organizations and should have enough knowledge to guide their customers through the development process. Last, customers need to be agile in order to deliver requirements and continuous feedback in an appropriate manner.

We conclude that some agile practices in the organization are not managed to be implemented in a good way or not implemented completely. The studied organization needs more training to get agile practices completely implemented.

The challenges related to generic RE are lack of domain knowledge and lack of experience of writing requirements. On the first hand, the lack of domain knowledge is a general RE problem and not related to agile. Moreover, lacking of knowledge can affect any development process. On the other hand, lack of experience of writing requirements causes lack of clarity of requirements which causes wasting time to get understand the requirements. To solve these challenges, the employees need more training in order to increase their knowledge about RE [45].

Bibliography

[1] Minna Pikkarainen, Jukka Haikara, Outi Salo, Pekka Abrahamsson, and Jari Still. The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3):303–337, 2008.

[2] James A Highsmith. *Agile software development ecosystems*, volume 13. Addison-Wesley Professional, 2002.

[3] Orit Hazzan and Yael Dubinsky. Why software engineering programs should teach agile software development. *ACM SIGSOFT Software Engineering Notes*, 32(2):1–3, 2007.

[4] David Cohen, Mikael Lindvall, and Patricia Costa. An introduction to agile methods. *Advances in computers*, 62(03):1–66, 2004.

[5] Mark Paulk. Capability maturity model for software. *Encyclopedia of Software Engineering*, 1993.

[6] Alberto Sillitti and Giancarlo Succi. Requirements engineering for agile methods. In *Engineering and Managing Software Requirements*, pages 309–326. Springer, 2005.

[7] Ian K Bray. *An introduction to requirements engineering*. Pearson Education, 2002.

[8] Frauke Paetsch, Armin Eberlein, and Frank Maurer. Requirements engineering and agile software development. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on*, pages 308–313. IEEE, 2003.

[9] Balasubramaniam Ramesh, Lan Cao, and Richard Baskerville. Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, 20(5):449–480, 2010.

[10] Ville T Heikkilä, Daniela Damian, Casper Lassenius, and Maria Paasivaara. A mapping study on requirements engineering in agile software development. In *Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on*, pages 199–207. IEEE, 2015.

[11] Murali Chemuturi. *Requirements engineering and management for software development projects*. Springer Science & Business Media, 2012.

[12] Ian Sommerville and Pete Sawyer. *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., 1997.

[13] Irum Inayat, Lauriane Moraes, Maya Daneva, and Siti Salwah Salim. A reflection on agile requirements engineering: solutions brought and challenges posed. In *Scientific Workshop Proceedings of the XP2015*, page 6. ACM, 2015.

[14] Elizabeth Bjarnason, Krzysztof Wnuk, and Björn Regnell. A case study on benefits and side-effects of agile practices in large-scale requirements engineering. In *Proceedings of the 1st Workshop on Agile Requirements Engineering*, page 3. ACM, 2011.

[15] Dan Turk, Robert France, and Bernhard Rumpe. Limitations of agile software processes. *arXiv preprint arXiv:1409.6600*, 2014.

[16] Mikko Korkala, Pekka Abrahamsson, and Pekka Kyllonen. A case study on the impact of customer communication on defects in agile software development. In *Agile Conference, 2006*, pages 11–pp. IEEE, 2006.

[17] Linda A Macaulay. *Requirements engineering*. Springer Science & Business Media, 2012.

[18] Grischa Liebel, Matthias Tichy, Eric Knauss, Oscar Ljungkrantz, and Gerald Stieglbauer. Organisation and communication problems in automotive requirements engineering. *Requirements Engineering*, pages 1–23, 2016.

[19] Tim Goles and Wynne W Chin. Information systems outsourcing relationship factors: detailed conceptualization and initial evidence. *ACM SIGMIS Database*, 36(4):47–67, 2005.

[20] Maryam Alavi and Dorothy E Leidner. Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS quarterly*, pages 107–136, 2001.

[21] Judith Weedman. Informal and formal channels in boundary-spanning communication. *Journal of the American Society for Information Science*, 43(3):257, 1992.

[22] NSAE Kandlousi, Anees Janee Ali, and Anahita Abdollahi. Organizational citizenship behavior in concern of communication satisfaction: The role of the formal and informal communication. *International Journal of Business and Management*, 5(10):51–61, 2010.

[23] Jeanette W Gilsdorf. Organizational rules on communicating: How employees are-and are not-learning the ropes. *Journal of Business Communication*, 35(2): 173–201, 1998.

[24] Judy Gray and Heather Laidlaw. Part-time employment and communication satisfaction in an australian retail organisation. *Employee Relations*, 24(2): 211–228, 2002.

[25] Ken Schwaber. Scrum development process. In *Business Object Design and Implementation*, pages 117–134. Springer, 1997.

[26] Ken Schwaber. *Agile project management with Scrum.* Microsoft press, 2004.

[27] Artem Marchenko and Pekka Abrahamsson. Scrum in a multiproject environment: An ethnographically-inspired case study on the adoption challenges. In *Agile, 2008. AGILE'08. Conference*, pages 15–26. IEEE, 2008.

[28] Jeffrey K Liker. *The toyota way.* Esensi, 2004.

[29] Muhammad Ovais Ahmad, Jouni Markkula, and Markku Oivo. Kanban in software development: A systematic literature review. In *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*, pages 9–16. IEEE, 2013.

[30] David J Anderson. *Kanban: successful evolutionary change for your technology business.* Blue Hole Press, 2010.

[31] Ozden Ozcan-Top and Onur Demirörs. Assessment of agile maturity models: a multiple case study. In *International Conference on Software Process Improvement and Capability Determination*, pages 130–141. Springer, 2013.

[32] Chetankumar Patel and Muthu Ramachandran. Agile maturity model (amm): A software process improvement framework for agile software development practices. *International Journal of Software Engineering, IJSE*, 2(1):3–28, 2009.

[33] Arbetsförmedlingen. What is arbetsformedlingen, 2016. URL `http://www.arbetsformedlingen.se/Globalmeny/Other-languages/AWhatisarbetsformedlingeboutus.html`. [Online; accessed 15-Sept-2016].

[34] Atlassian. Jira software. URL `https://www.atlassian.com/software/jira`.

[35] ReQtest. We are reqtest. URL `http://reqtest.com/about-us/`.

[36] Usabilla. About usabilla. URL `https://usabilla.com/about`.

[37] Carolyn B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering*, 25(4):557–572, 1999.

[38] Per Runeson, Martin Host, Austen Rainer, and Bjorn Regnell. *Case study research in software engineering: Guidelines and examples.* John Wiley & Sons, 2012.

[39] Colin Robson. Real world research. 2nd. *Edition. Blackwell Publishing. Malden*, 2002.

[40] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*, pages 285–311. Springer, 2008.

[41] Timothy C Lethbridge, Susan Elliott Sim, and Janice Singer. Studying software engineers: Data collection techniques for software field studies. *Empirical software engineering*, 10(3):311–341, 2005.

[42] EG Guba and YS Lincoln. Naturalistic inquiry (vol. 75). *Beverly Hills, CA: Sage*, 1985.

[43] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[44] Joseph Maxwell. Understanding and validity in qualitative research. *Harvard educational review*, 62(3):279–301, 1992.

[45] Neil AM Maiden, Cornelius Ncube, and Andrew Moore. Lessons learned during requirements acquisition for cots systems. *Communications of the ACM*, 40(12): 21–25, 1997.

[46] Philipp Diebold and Marc Dahlem. Agile practices in practice: a mapping study. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, page 30. ACM, 2014.

[47] Arbetsförmedlingen. Arbetsförmedlingens återrapportering, 2013. URL `http://www.arbetsformedlingen.se/download/18.3485b9a713b6ad32ce18455/Redovisning_av_milj%C3%B6ledningsarbete_20121.pdf`. [Online; accessed 14-May-2017].

[48] Medieakademin. Förtroendebarometer, 2017. URL `http://medieakademien.se/wp-content/uploads/2017/04/Fortroendebarometern2017.pdf`. [Online; accessed 15-May-2017].

[49] Talarforum i Skandinavien AB. Why swedes are mad about meetings, 2017. URL `https://www.talarforum.se/artikel/colin.moon/223/223/why_swedes_are_mad_about_meetings.html`. [Online; accessed 09-May-2017].

A

Appendix I

Interview Questions with Internal Stakeholders

1. Which role do you have in the organization?
2. How long have you worked in this role?
3. What agile practices do you use?
4. Which product are you working with? (in case they just say the name of the product). So maybe prepare some follow-up questions (how many developers?
5. How is requirements engineering done in your team for a specific product?
   (a) Where do requirements come from? What form? Tools?
   (b) Do you think the tool fails to address few issues? What are they?
   (c) What do you do with the requirements received? Refine/break them down?
   (d) Do you reuse requirements? How?
   (e) How do you manage changes in requirements?
   (f) How do you manage unclear requirements? How is it clarified? Make assumptions?
   (g) What challenges and benefits do you encounter with the current requirement process? How can challenges overcome?
   (h) What benefits and challenges exist while working together with requirements and user stories? How can challenges overcome?
6. What benefits of agile in AF work are? in case you know what happened before using agile.
7. Which challenges exist through using agile in the organization?
8. What kind of customers has AF?
9. Which communications channels use AF to communicate with those customers?
10. How much of informal communication do you rely on?
11. Which problems in communication with customers appear?
12. Which changes are planned to overcome those problems?
13. Which aspects of communication with customers work well?2
14. How would the ideal picture look like?

B

Appendix II

Interview Questions with Customers

1. How do you come up with your requirements?
2. Have you any contact with external stakeholders?
3. Which background do you have about Agile?
4. How do you communicate with the IT department?
5. How often provide you feedback for delivered issue?
6. Is there any problem related to the communication with stakeholders?

C
Appendix III

Interview Questions with External Stakeholders

1. How you provide your feedback?
2. Which relation do you have with AF?
3. Have you any contact with the IT department?
4. Is there any challenge related to communication with the IT department?