The Effects of Uncertainties in Partially Distributed Agile Software Development Teams

REBECCA KLINC
The Effects of Uncertainties in Partially Distributed Agile Software Development Teams

Rebecca Kline
Effekterna av Osäkerheter i Delvis
Distribuerade Agila
Mjukvaruutvecklingsteam

Rebecca Kline
Abstract

**Purpose:** The purpose of this study is to identify how uncertainties affect a partially distributed team. To do this, the thesis goes into the details of the different types of uncertainties and how they are handled.

**Design/Methodology/Approach:** This research uses a multiple case study with interviews and observations as data gathering methods. The two cases study are first analyzed separately and then a cross case analysis is conducted.

**Findings:** This study showed that uncertainties are mostly correlated to threats and therefore focus lies on mitigating them. Using agile practices allows for greater flexibility and makes uncertainties more manageable.

**Practical Implications:** This study shows the value of following agile principles. It also shows that when choosing to have a partially distributed team one needs to weigh the negative aspects with the positive ones.

**Originality/Value:** This study combines research conducted on partially distributed teams and uncertainties.

**Paper Type:** Master thesis

**Keywords:** Uncertainty, Partially distributed teams, Disciplinary agile delivery, Agile software teams, Software development, Project management
Sammanfattning

Syfte: Syftet med denna studie är att identifiera hur osäkerheter påverkar ett delvis distribuerat team. För att göra detta går uppsatsen in i detalj om olika typer av osäkerheter och hur de kan hanteras.

Design/Metodologi/Ansats: Denna studie använder sig av fallstudier med intervjuer och observationer som datainsamlingsmetoder. Två fallstudier analyseras först separat och därefter genomförs en korsfallsanalys.

Resultat: Denna studie visade att osäkerheter mestadels är korrelerade med hot och därför ligger fokus på att förhindra dem. Genom att använda agila metoder möjliggörs större flexibilitet och gör osäkerheter mer hanterbara.

Praktisk betydelse: Denna studie visar värdet av att följa agila principer. Det visar också att när man väljer att ha ett delvis distribuerat team måste man våga de negativa aspekterna med de positiva.

Originalitet/Värde: Denna studie kombinerar forskning som utförs på delvis distribuerade team och osäkerheter.

Rapporttyp: Examensarbete

Nyckelord: Osäkerhet, Delvis distribuerade team, Disciplinary agile delivery, Agila mjukvaruteam, Mjukvaruutveckling, Projektledning
Contents

**Foreward** vii

1 Introduction 1

1.1 Background 1
1.2 Problem Description 3
1.3 Purpose 4
1.4 Scientific Questions 4
1.5 Delimitation and Limitations 4
1.6 Ethics 4
1.7 Scientific Contribution 5
1.8 Outline 5

2 Disciplined Agile Delivery 7

2.1 Life-Cycle 7
2.2 Primary Roles 8

3 Literature Review 10

3.1 Agile Software Development 10
3.2 Lean Software Development 11
3.3 Uncertainty in Relation to Risk 12
3.4 Uncertainty in Agile Development 12
3.4.1 Market Uncertainty 13
3.4.2 Requirement Uncertainty 13
3.4.3 Socio-Human Uncertainty 14
3.4.4 Resource Uncertainty 14
3.4.5 Environmental Uncertainty 15
3.4.6 Task Uncertainty 15
3.4.7 Technological Uncertainty 16
3.5 Partially Distributed Teams 16
3.5.1 Tools & Communication 16
3.5.2 Group dynamic 17
3.5.3 Management & Practices 18

4 Methodology 20

4.1 Research Design 20
4.2 Literature Review 21
4.3 Data Gathering 21
4.3.1 Interviews 21
4.3.2 Pilot Study 22
4.3.3 Multiple Case Studies 23
4.4 Data Analysis 25
4.4.1 Pilot Study 25
4.4.2 Multiple Case Studies 25
4.5 Validity & Reliability 26

5 Empirical Result 28

5.1 Context 28
List of Figures

1. Number of publications on agile practices on the ISI Web of Science ........ 2
2. Gantt Schedule of Master Thesis period ........................................ 21
3. Data Analysis ............................................................................. 25
4. Frequency of threats and opportunities mentioned ............................. 48
5. Number of uncertainties identified .................................................. 49

List of Tables

1. Overview of conducted introductory interview ..................................... 22
2. Overview of conducted navigational interviews .................................... 23
3. Overview of conducted interview for Project 1 .................................... 24
4. Overview of conducted interview for Project 2 .................................... 24
5. Overview of uncertainties and sources of uncertainties ....................... 32
6. Overview of uncertainties and sources of uncertainties ....................... 35
7. Overview of identified uncertainties for Project 1 ............................... 40
8. Overview of identified uncertainties for Project 2 ............................... 45
9. Overview of how uncertainties are managed ...................................... 52
Foreward

This research was conducted as my master thesis and is part of my education at the institution for Industrial Engineering and Management at the Royal Institute of Technology (KTH) in Stockholm. It is a course corresponding to 30 ECTS and is the final assignment of five years of studies. The study was conducted between December of 2017 and June of 2018.

I would like to begin by thanking Dewire and all employees for welcoming me to conduct my master thesis there. A special thanks to my supervisor Petter Andersson who have helped me find people to interview and helping me with the report. I would like to give an extra thanks to all interviewees who have taken time from their tasks in order to answer all my questions.

I would also like to thank my supervisor at KTH, Lars Uppvall who have helped me steer this research in an appropriate direction. He have helped me rethink and motivate all choices that has been done along the way.

Finally I would like to thank all people who have read the different drafts of my thesis and given valuable feedback, among these people we have: master thesis students at Dewire, my opponent at KTH and family members. These people helped me raise the standard of the thesis.

Rebecca Klinic
Stockholm, June 2018
1 Introduction

Introductory chapter to familiarizes the reader with the topic. The background of the study is described initially, followed by the purpose and the scientific questions. Also, the delimitation and the limitations of the research are discussed as well as the scientific contribution of the study. At the end of the chapter, the outline of the thesis is laid out.

1.1 Background

In today’s global focus on the concept of digitalization have resulted in the use of geographically distributed teams becoming more common. Such teams communicate through emails, calls, video conferences and virtual status boards. However, information is lost or not available to all members within a partially distributed team, due to the level of communication varying between the local and remote team members. [Webster and Wong, 2008]. Having people in different locations may ease communication with customers and it is also possible to gain a greater resource pool [Conchaír et al., 2005]. This results in a feeling of having access to more expertise than a co-located group [Webster and Wong, 2008].

As a result of digitalization IT projects has become of great importance for industries. Still, roughly 1/3 of all IT projects fail [The Standish Group Report, 2014]. The report by [The Standish Group Report, 2014] identified the three main reasons for project failure as:

1. Lack of user input
2. Incomplete requirements and specifications
3. Changing requirements and specifications.

Different management methods can be applied depending on a project’s characteristics. These management styles include the traditional waterfall-, agile-, iterative- and lean methods. However, the complexity and success rate vary between these methods and some might not be suitable for certain types of projects.

The waterfall management style focuses on a detailed overhead plan in which the different steps are completed in a specific order. All the programming is done first, then testing and if it is approved, then the entire project is launched or given to the customer. However, due to long project time, a rapidly changing environment and poor project specification these projects can result in something which the customer does not need. The waterfall approach may lead to bottle necks in the project because testing is done in the end. Therefor errors might not be found until a few days before the release day. These issues result in a low success rate. A survey conducted by [Ambysoft, 2013] reported that only 49% of all IT projects using this approach succeeds.

The concept of agile software development was conceived at the end of the 20th century and started gaining attention around the turn of the century. In 2001, a group of practitioners published the Manifesto for Agile Software Development, also known as the Agile Manifesto, in which they stated that they in their work have come to value [Beedle et al., 2001]:

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

In essence, agile software development is characterized by a rejection of traditional methods of careful planning and sequential execution in favor of a more dynamic and flexible approach (Dingsøyr and Dybå, 2008). It is on these values which agile methods are based on (Balijepally et al., 2012). Agile Software Development is at its core flexible in order to manage uncertainties and new demands and works in iterations which allow a incremental and/or iterative process of the tasks. The survey by Ambysoft (2013) also reported that only 6% of the IT projects using an agile method failed, making them a popular choice among IT teams.

The flexibility of agile methods can in turn result in great opportunities for software teams. Agile methodology focuses on turning the requirements into smaller tasks for each iteration, thus creating small components which can be re-used later.

Figure 1: Number of publications on agile practices on the ISI Web of Science.

Before the release of the Agile Manifesto the more traditional waterfall approach was used for software projects. After the Agile Manifesto more publications regarding agile methods and their principles were published and at the same time more teams began using an agile method, see figure 1 (Balijepally et al., 2012). With the growing interest in agile methods different versions of it have been examined and used in practice, such as Scrum, eXtreme Programming (XP), Lean software development and Disciplined Agile Delivery, (DAD) (Balijepally et al., 2012). Still, with high focus on everyday work and the upcoming delivery, it can be challenging to dodge threats and exploit opportunities when they arise.

This paper will focus on the agile method DAD, which is the main method used by this papers
commissioner, Dewire. Dewire is a medium sized, roughly 70 employees, IT-consultancy firm with offices in Stockholm and Sundsvall. A majority of the employees are located in Sundsvall, while the rest are located in Stockholm. Since Dewire has employees at two different geographical location, some teams are partially distributed. Projects taken by Dewire are both in-house projects, where Dewire sets up their own teams and projects where consultants work for an external organization. In regards of the latter Dewire can not affect the development method, since it is decided by the customer; in regards to the former Dewire can use a method of their choice, but is affected by the customers budget, experience, time frame and possibility to be involved.

1.2 Problem Description

For software development projects uncertainties is to be expected (Balijepally et al., 2012). Iteration lengths are usually accustomed to fit with the feedback rate from the customer. However, customers may still come with projects changes in the middle of an iteration. A problem with many agile methods is that they have misinterpreted the Agile Manifesto and minimize the documentation to a point where knowledge becomes implicit (Balijepally et al., 2012). As with any projects there might be a change in resources needed, thus leading to a decrease or increase of team members.

Still more uncertainties exist than those mentioned, hence the name uncertainty. It is important to note that uncertainties include opportunities as well as threats. Usually IT teams do handle threats in a standardized way, either subconsciously or consciously. Opportunities are usually handled based on the initiative of an individual team member. The way of handling threats is usually a mix of different agile methods, which a team has learned through experience (Dönmez and Grote, 2018). This shows that no standardized method of handling threats is established and at the same time several opportunities go unnoticed due to a lack of knowledge of how to identify them. When working with agile development in partially distributed teams, one needs to take four things into consideration:

1. Calendar time of the project: In practice this would mean that planning for uncertainty is made both before the project, and maybe during every iteration planning and iteration retrospective. For this a tool or framework is needed to find and identify which areas of uncertainty to which the project is vulnerable to.

2. Timing: How uncertainties are identified and handled during the projects is essential. The timing of an uncertainty should not affect the project. A well performing team should be able to identify both threats and opportunities mid-iteration.

3. Proof of concept: When working with new technology it would be beneficial to work with proof of concept in order to estimate implementation time and learning lessons for similar projects in the future, thus turning tacit knowledge into explicit.

4. Geographical distribution: With the help of new technology it is possible for a team to be distributed over various locations. This type of distribution tends to create an ”Us vs Them” feeling which can lead to teams being more competitive rather than co-operative (Hiltz et al., 2013), but it also allows a more varied resource pool.

The agile methodology is complex and requires active participation from all team members. The multiple dimensions of agile methodology is what allows for flexibility to handle uncertainty. Still
teams have a hard time finding which tools to use depending on the type of project. At its core agility suits smaller co-located teams better since it opts for transparency and shared leadership within the team. However, due to the increase of partially distributed teams even more uncertainty arise, which stem from the geographical distribution.

1.3 Purpose

The purpose of this thesis is to identify how uncertainties affect a partially distributed agile software development team. In order to fulfill the purpose, the uncertainties which partially distributed agile software development teams are subject to need to be identified; as a final step one needs to identify how these uncertainties are being handled.

1.4 Scientific Questions

The following main research question (MRQ) have been identified to fulfill the purpose.

- How are partially distributed agile software development teams affected by uncertainties?

In order to answer the MRQ two sub research questions (RQ) have been developed.

- RQ1. What areas of uncertainty are partially distributed agile software development teams typically subject to?
- RQ2. How do partially distributed agile software development teams handle uncertainties?

1.5 Delimitation and Limitations

I have chosen to limit the study to focus on one company using the DAD framework. Another aspect I have chosen to limit this study on is geographical distance. The teams are fairly close in distance, a 3,5 hour train journey, making it possible for team members to commute if needed. However this geographical distance also limits my possibility to spend equal time observing both offices, hence the observations will be focused on the Stockholm office.

The limited time available makes it impossible for me to follow an entire project from start to finish, which means that through observation I will not be able to notice all types of uncertainties. Even uncertainties which occur later on in the project which might be caused by poor project initialization will be hard for me to analyze. Due to customer discretion and secrecy, uncertainties based on poor customer decisions might not be as easy to identify, thus limiting the uncertainties which can be found.

1.6 Ethics

The thesis was conducted at an IT consultancy firm, thus two parts of the ethical aspect should be considered. It needs to be ensured that the thesis is ethical in regards of handling company data as
well as in regards to the individuals being observed and interviewed at the company. The former is related to information about clients and their projects. Due to this, any description or mention of a project or customer in the thesis has been censored. Then before being added to the thesis it was read and approved by an employee at the case company. Before starting the thesis the researcher had to sign a confidentiality agreement, thus forcing confidentiality at a company level.

In regards of the latter, the ethical principles described by Bell and Bryman (2007) have been followed. These principles are: no harm to participants, dignity, informed consent, privacy, confidentiality, anonymity, deception, affiliation, honesty and transparency, reciprocity and no misrepresentation. Before each interview officially started the participants were informed of the purpose of the research and the interview. They were also made aware of how their answers and persona would be displayed in the Methodology and Empirical Result chapter before asking for consent of recording and conducting the interview. The recordings and transcribed notes are only available to the author of the research. After the research is completed the results will be presented to the case company, hence they will benefit from it which will hopefully reduce misleading answers from the participants.

1.7 Scientific Contribution

From the literature study several articles discussing the different aspects of agile methods exist. More recent articles also discusses the aspect of risk management and how to integrate them depending on the team structure. However, these are mostly in regards to known possible threats. When discussing uncertainties in a neutral context, both threats and opportunities, very few articles have been found and none have focused on partially distributed teams. This gives a great opportunity for this research to contribute with how uncertainties affect partially distributed agile software development teams.

1.8 Outline

In chapter 1 the reader is provided with the relevant background and a description of the problem. Then the purpose and the research question of the thesis are presented. This is followed by a discussion regarding delimitation, limitations and scientific contribution. This chapter ends with the outline of the thesis.

In chapter 2 a description of Disciplinary Agile Delivery is given. The chapter presents the different phases in the development process as well as the primary roles needed. This is relevant for the understanding of the case, as it supposedly is the agile method used by the case company being studied.

In chapter 3 a summary of the literature study is presented to the reader. This is to give the reader an understanding on the basis for the thesis and an opportunity to read up on subjects which they might find interesting.

In chapter 4 the method of the research is described. Relevant literature has been researched and translated into the method which will be used to fulfill the purpose of this study. The chapter describes how the empirical data is gathered and analyzed. It ends with a section describing how validity and reliability are ensured.
In chapter 5 the reader is presented with the context and a description of the two cases being studied. This is followed by the empirical results from the pilot study and two case studies in order to make an analysis possible. The result from the two different cases are presented individually.

In chapter 6 the results are analyzed based on the literature study and later on discussed. The two cases studies are first analyzed separately, followed by a cross case analysis and a discussion presenting answers to the research questions.

In chapter 7 a conclusion is given stating the answers to the research questions. This is followed by practical and theoretical implications of this study. The chapter ends with a section on possible future research.
2 Disciplined Agile Delivery

This chapter will describe the Disciplined Agile Delivery framework as described by Ambler and Lines (2012) in their book “Disciplined Agile Delivery: A Practitioner’s Guide to Agile Software Delivery in the Enterprise”. Hence this chapter will describe the original DAD method.

Ambler and Lines (2012) believe that agile methods has taken on an extreme form with the only goal being to create working software. Common agile methods such as eXtreme Programming (XP) and scrum focuses on the construction phase, thus making them good for smaller projects. However, larger and more complex projects needs an established guide for the initial inception phase and the final transition phase. DAD is a hybrid method, derived from common agile methods such as scrum, XP, Unified Process and Agile Modelling, just to mention a few.

Like most agile methods DAD is based on the Agile Manifesto, however they differ on some points. The principle: "Working software over extensive documentation" is one which DAD does not agree to. The idea behind the method is to have a working solution rather than software, and for that documentation is needed. The principle "customer collaboration over contract negotiation" is not optimal since it does not take all stakeholders into account. DAD points out the importance to consider the input of all stakeholders and not just the ones who pay.

The DAD framework is based on common agile methods, but it also uses lots of lean principles. This allows DAD to be more scaleable than other agile methods. The authors explain how lean principles clearly gives the logic to why agile methods usually succeed. See section 3.2 for a description of lean software development.

2.1 Life-Cycle

A special aspect to the DAD method is that it has three different phases in order to make the process from start to finish smoother. During the project continuous goals exist, such as the growth of all team members, addressing risks, improving processes and infrastructure.

Inception Phase: During this phase the team is formed with the goal of finding a vision and getting it approved by the stakeholders. Initial technological structure needs to be determined and possible risks should be identified. This phase usually takes place between a few days up to a few weeks. It is important to use relevant tools in order to get the team on the same page, a focus should be on visualization techniques as it gives a clear overview.

Construction phase: During this phase the solution is created and the team needs to be aware of the change in demand from the stakeholders. The construction phase consists of short iterations with incremental implementation. Within the first few iterations, the architecture needs to be proved in order to gain trust from the stakeholders. The lengths of these iterations are between one and four weeks. Each iteration consists of an iteration planning and modelling. At the end of an iteration a demo is presented to the stakeholders and a retrospective is held with the team members. This phase is implemented using a Risk-Value driven priority. This results in prioritizing tasks based on stakeholder requirements and possible risk. By implementing high risk tasks early it is possible to see which direction a project might take and it will also be possible to spot project “fails” at an earlier stage.
**Transition phase:** During this phase the product will most likely already have been completed. However, the project still needs a few finishing steps. Before the end of a project the stakeholders should be ready to receive the product and the product solutions should have been deployed. It is not until the solution has been used for a few days and the stakeholders are pleased that a DAD team can consider the project done.

### 2.2 Primary Roles

Within the DAD framework it is important to remember that the roles are not positions and they are flexible, meaning that one person can take on different roles for different projects. For smaller projects it might also be possible for one person to take on two roles.

**Team Member:** The team members are the people who create the solution for the stakeholders. The DAD methodology believes that all team members should possess multiple skill sets, but not master them all. By having team members with multiple skill sets you reduce resource uncertainty. Team members are supposed to identify tasks, estimate completion time and perform the tasks. The team lead and architect owner are also considered to be team members.

**Team Lead:** The goal of the team lead is to be an agile coach for the other team members. Other responsibilities include: assessing the team members, managing the project budget and make sure that relevant documentation exists. It is not within their responsibilities to make decisions, but they should facilitate decision making and act as moderator. [Ambler and Lines] (2012) mention that the team lead is similar to a scrum master, but since they do not want to favor a specific agile method they use the more neutral term: team lead. The team lead should not be confused with a team manager.

**Architect Owner:** The architect owner creates the overall design of the solution and lets it evolve during the process. It is beneficial to have a more senior employee at this role, someone who knows what subsystem, frameworks and patterns that are used by the company.

**Product Owner:** The product owner is supposed to represent the stakeholder. It is his/her job to make sure that the team and the stakeholders are aware of the requirements. In practice this means that whenever a requirement uncertainty arises the product owner is supposed to resolve it or find the information needed to resolve it. The product owner is the communication link between the team and the stakeholders. The product owner is a vital part of the development process. However, a product owner faces multiple challenges:

- When working with the same team constellation over a period of time and become more aware of how they work, it is possible that this impact the relationship with the stakeholders negative.
- Since the product owner is the link between the team and stakeholder he/she can easily become the bottle neck of a project, especially when a person is the product owner of separate projects.
- The importance of the role means that if the product owner would have to leave the project, it can hinder the development since the communication link disappears.
- Due to the large number of stakeholders, it is usually impossible to represent all of them.
- If the product owner for a larger team is not a skilled negotiator it can result in an imbalance among the sub-teams distributions.

**Stakeholders:** An aspect which differs DAD from many other agile methods is the use of the
term stakeholder, rather then client. DAD recognizes that using the client for feedback might not be optimal due to the fact that other stakeholders, such as end users or governmental exists which might affect the spread of the product/project. When developing a product it is important to identify all stakeholders and fit the product according to their needs and requests. Within the DAD framework four stakeholders have been identified:

- **End User:** They are the ones who are going to use the product.
- **Principals:** The decision maker of the end users, usually have a senior management role. They are the client of the project, since they make the payment arrangements.
- **Partners:** These are the people who help integrate the system with the client organization. They are usually support and operations staff, legal experts, application developers.
- **Insiders:** These are the people helping with your project and business. They are team members, suppliers of technical solutions.
3 Literature Review

In this chapter the result of the literature review is described. Initially relevant frameworks are described and discussed. This is followed by a section on research on uncertainty in relation to risk and common areas of uncertainties. The chapter ends with theory on geographically distributed team and their group dynamic.

3.1 Agile Software Development

One of the core principles of the Agile Manifesto (Beedle et al., 2001) is to add value to the project and deliver it on time. Agile software teams defines value aspects as delivery process w.r.t time, perceived quality, cost, actual quality and processes, ways of working and tool (Alahyari et al., 2017). Westerveld (2003) identified three criteria of success: budget, schedule and required quality which can be translated to the four first value aspects of agile software teams. This shows that goals of agile software teams do not differ from traditional project goals, but other terms and methods are used to achieve the goals.

Agile software development differs from the traditional approach in order to be more flexible. However, this also leads teams to being more vulnerable to certain risks and uncertainties. The goal of delivering after every iteration can in some cases result in a stressed team which in turn take short cuts to finish within the iteration (Elbanna and Sarker, 2016). When working with agile methods an experienced team should be able to make calculated estimations of completion time, which allows them to complete all tasks within an iteration. Agile software teams are self-organizing giving freedom of choice in regards of technology and testing. This freedom also results in unstandardized methods, which can result in conflict when joining an ongoing project or creating a new project team (Elbanna and Sarker, 2016). The flexibility associated with agile methods also leads to uncertainty when making decisions, hence they might be focused on short term gains (Acton et al., 2017).

To decrease bureaucracy and wasteful activities agile methods advocate less documentation than the waterfall approach. The decrease in documentation makes the on boarding process for new team members more difficult and it also turn knowledge implicit rather than explicit. By having implicit knowledge within a team, the project becomes more dependant on the team members, thus a loss of a team member will be hard to cover. This claim is strengthened by Elbanna and Sarker (2016) who claim that projects tend to decrease their efficiency when a new member joined the team. Another downside of not valuing documentation is that if not all decisions are being documented, it can be hard, or sometimes impossible to plan since there is data missing (Acton et al., 2017).

The four agile values, if not implemented with caution, can result in reduced efficiency rather than increased flexibility. When individuals are valued higher than processes it can result in more experienced team members dominating the decision making processes and past problems can be repeated (Acton et al., 2017). However, this might also be the result of how team members might be unwilling to make a decision during times of high uncertainty (Conboy et al., 2012).

Tools for managing communication and knowledge sharing in agile projects include sprint planning, retrospectives and daily stand ups, to mention a few (Ambler and Lines, 2012). The most commonly used agile method Scrum conducts sprint planning by having the product owner, team lead and team...
members review the highest prioritized tasks and setup up a plan for the coming sprint (Cooke, 2012). In order to plan the coming sprint the team needs to estimate the completion time of the different tasks and task dependencies needs to be clear. Retrospectives are held after each sprint as a way to enable continuous learning (Dingsøyr, 2005). The three steps of retrospectives are (Lehtinen et al., 2017):

1. Target definition: this step aims on choosing a topic for the retrospective.
2. Reflection: this step aims to identify experiences during the iterations, hence it can be used to identify opportunities and threats.
3. Corrective action development: this step aims to find actions which can improve the project.

3.2 Lean Software Development

Lean software development is what gives DAD its competitive edge compared to other agile methods (Ambler and Lines, 2012). Poppendieck (2007) argues that the success from agile methods can be explained by lean software development. Poppendieck and Poppendieck (2003) have based their framework on the lean principles:

1. **Eliminate waste:** Waste is considered as anything which does not add value to the project. However, it should be noted that this does not mean scrap documentation. When working with a project a team should be responsible for it and a task should not be handed around as this reduces efficiency.
2. **Amplify learning:** During the process of software development each iteration should increase the knowledge of the software. The final product is not determined in the early sprints, it should evolve as the customer gets to see different variation of it and give feedback.
3. **Decide as late as possible:** Due to the flexibility which agile and lean methods advocate to make decision as late as possible. This does not mean to procrastinate decision, but that in times of great uncertainty it might be beneficial to make decision once more information exists.
4. **Deliver as fast as possible:** By delivering as fast as possible it allows for receiving feedback earlier. This allows teams to design, implement and getting feedback within one sprint, thus improvements can be made early on.
5. **Empower the team:** All members of the team have their specialty and therefore all should be a part of strategical decisions and planning. The developers are the people with detailed knowledge of certain parts. Thus, they can help with architectural design and decisions.
6. **Build integrity in:** The developed software is supposed to have high integrity as this allows for long term success. It should be possible to add new features and upgrade the solution if needed. The software should score high in usability test and when working with commercial software it should have a high market share.
7. **See the whole:** Team members with multiple skill sets can increase the value of the software. Since they can get a more holistic view of the software and architectural decisions they can help increase the overall performance.

A big difference between classic agile development and lean software development is the role of product owner and customer. Lean development does not have a product owner or customer role as it violates the principle of "see the whole" (Cusumano and Poppendieck, 2012). The team should therefore strive as a whole that the product reaches the market and ensure its success. However, this might not be applicable for consultancy projects since their part in the products are limited to the development and then handed over to the customer. The product might even be one which is to
be used internally by another company, resulting in not being available on any commercial market.

3.3 Uncertainty in Relation to Risk

The definition of the term uncertainty within agile research is not clear. Different researchers used different definition. Dönmmez and Grote (2018) researched uncertainty as a neutral concept while Rathod and Shrivastava (2015) uses the term uncertainty interchangeably with risk. The definition of the term uncertainty is of great importance since it affects how the uncertainty is handled. A clear example is in the research Managing uncertainty in software projects (Marinho et al., 2017) where the focus lies on reducing uncertainties through early detection. Hence, only focusing on threats and not even mentioning opportunities.

According to Oxford University Press (2018) risk is defined as :

"The possibility that something unpleasant or unwelcome will happen."

and uncertainty is defined as:

"A state which is not known or definite."

The definitions above corresponds to how Dönmmez and Grote (2018) and Gustafsson et al. (2008) defines it. When working with agile methods, the planning period is supposed to be short, thus leaving room for uncertainties. Hence the term is very common within agile software teams. Through out this thesis uncertainty in relation to software development will be defined as "A state or an outcome which is not known or definite", hence leaving room for both positive and negative outcomes or states, also known as opportunities and threats.

3.4 Uncertainty in Agile Development

The research on uncertainties is very broad. A few articles refer to uncertainties, but under close inspection they have been shown to use the term uncertainty as a synonym for risk (Hillson, 2002; Rathod and Shrivastava, 2015). All uncertainties mentioned in this thesis are based on the definition in section 3.3. The main types of uncertainties which have been found are market, requirement, socio-human, resource, environmental, task and technological. They have been identified from different articles, hence the definition of some might overlap in some areas and a few uncertainties within one area might result in uncertainties in other areas.

There are two possible ways to handle uncertainties, by mitigating them or learning how to cope with them and exploit them (Grote, 2004). By mitigating uncertainties organization will lose the opportunities which they stem from. An increase in flexibility reduces the impact of uncertainties and it allows teams to exploit the opportunities which may stem from uncertainties (Lima et al., 2014). It is important that all team members can handle uncertainties locally and not just on a managerial level (Grote, 2004).
3.4.1 Market Uncertainty

Market uncertainty relates to uncertainty regarding how the market perceives the product (Marinho et al., 2017). For products which the market is familiar to it is possible to use previous figures to estimate sales and profits. If the product is new to the market then the market uncertainty is high. Furthermore, if rapid changes to the market are ignored it can lead to a reduction in sales (Jaafari, 2001). Early identification of market changes can lead to the mitigation of threats and the possibility to exploit opportunities.

Market uncertainty is not project specific, but rather pre-determined by the market (Beckman et al., 2004). Beckman et al. (2004) also found that during periods of high market uncertainty a firm usually strengthen their alliances with partners and expand their network. By creating stronger alliances a firm can ensure that they have sufficient suppliers or buyers in order to create a product, thus reducing threats which stem from market uncertainty. Due to market shifts a project team should always be aware of new segments which should be targeted (Jaafari, 2001). If ignoring market shifts teams can turn the ignorance of an uncertainty to a threat, rather than exploiting the opportunity while it is still possible.

3.4.2 Requirement Uncertainty

Requirement uncertainty is based on unclear and changing customer demands (Dönmez and Grote, 2013). A common conclusion from researchers and organizations (Dönmez and Grote, 2018; Lima et al., 2014; Rathod and Shrivastava, 2015; The Standish Group Report, 2014) are that many uncertainties arise from the ambiguous requirements from customers and stakeholders. The nature of agile methods is to allow flexibility, which makes it possible to address the swift changes to requirements from customers (Ceschi et al., 2005). However, this flexibility also allows the customer to change their demands, thus resulting in an unclear vision or goal for the project. According to The Standish Group Report (2014) requirement uncertainties which are turned into threats are among the most common reasons for IT projects failing.

Baskerville et al. (2013) found that with high levels of requirement uncertainty it is beneficial to use a more agile approach rather than a waterfall approach. This is to allow the flexibility which is needed to cope with changing requirements. It has been showed that through customer demo and user reviews that it is possible to mitigate the threats based on requirement uncertainty (Chan et al., 2008). By improving customer or stakeholder relationship it is possible for the development team to identify ambiguous requirement and efficiently get a clearer image of both specific requirements as well as the final vision of the solution (Ambler and Lines, 2012).

Requirement uncertainty usually shows itself through a lack of details surrounding a demanded task, ambiguous information about a requirement, unexpected changes to the requirements and new emerging requirement throughout the project (Dönmez and Grote, 2013). By following agile practices it is possible to avoid the downfalls which follows from the threats of these uncertainties. By shortening the time for an iteration it is possible to reduce the number of requirements changes which are added mid-iteration (Dönmez and Grote, 2018). By having a clear product owner who communicates with the stakeholders it is possible to get a clearer understanding of ambiguous requirements (Ambler and Lines, 2012; Beedle et al., 2001).
3.4.3 Socio-Human Uncertainty

Socio-human uncertainty are uncertainties which are related to knowledge sharing and the relationship between the members in a team (Marinho et al. 2017). The nature of the uncertainty results in being subjective between the members within a team, hence only the team itself can handle these uncertainties. Agile methodology requires communication and knowledge sharing. By organizing daily stand-ups communication between team members is unavoidable. However, it is up to all team members to handle socio-human uncertainty, not just the project manager and scrum master (Dvir and Shenhar 2007).

Socio-human uncertainties may stem from the team members themselves. Finding the optimal team members results in different team dynamic and levels of communication (Curşeu and Meslec 2015). By having a more diverse team it has been shown that you can create a more efficient team (Prichard and Stanton 1999). When having a diverse team the project manager can use each members strength when applicable and find people whose weakness are counteracted by another members strength. However, the possible team members might be affected by resource uncertainties.

3.4.4 Resource Uncertainty

Resource uncertainty stem from incomplete information regarding availability of all resources needed to finish the tasks, ranging from human resources to artifacts ( Dönmez and Grote 2013). It happens that resources are withdrawn, sometimes in the middle of an iteration (Conboy et al. 2012). These changes in resources affect an iteration greatly, especially since all tasks has an estimated completion time which would not be fulfilled. Due to time constraint resources pulled mid-iteration are hard to handle, but through follow up and planning it is possible to generate an improved plan for following iterations (Conboy et al. 2012).

Resource uncertainty shows itself through unavailable artifacts, unknown quality of input and the availability of human resources ( Dönmez and Grote 2013). Resource uncertainty might also in some cases be manifested through lack of information regarding knowledge within a team ( Dönmez and Grote 2018). The principle of pair programming, mostly used by XP-teams, are used in order to ease knowledge sharing within a team or organization, it allows for a natural spread of knowledge (Chan and Lui 2007; Ambler and Lines 2012). However, it is important to note the difference between lack of knowledge, an ambiguity, and lack of information, an uncertainty (Zack 2007). In regards of artifacts it is possible for a team to build basic technological infrastructure which can be used for other projects. By reusing old codes or code skeletons it is possible to exploit opportunities which arise.

Resource uncertainty also effects how efficiently a team works together. By not having adequate resources, such as part-time team members, lack of resources or an allocation of existing resource good team leadership is deterred (Dingsøyr and Lindsjøen 2013). However within a larger organization it is possible for teams to exploit opportunities in order to gain resources, both in form of finances and technology.
3.4.5 Environmental Uncertainty

Environmental uncertainty are uncertainties based on actions made by external actors as well as unpredictable changes to demographics and socio-cultural trends (Marinho et al., 2017). Environmental uncertainties are the result of a weakness in predicting external factors, however it should be noted that the external factors might turn into opportunities (Hoque, 2004).

Articles mentioning environmental uncertainty use the term *perceived environmental uncertainty*, this is because the level of uncertainty is not universal, but is set on an individual level (Downey, 1975). Waldman et al. (2001) states in their research that when environment uncertainty is perceived as high, it results in stress within the organization. The stress of environment uncertainty usually results in teams viewing the uncertainty as a risk, however a leader can use this to gain more trust (Waldman et al., 2001). Trust gained can then be used to create a new guiding vision for change, thus turning an uncertainty into an opportunity.

Waldman et al. (2001) found a correlation between how charisma and environment affected net profit margin. A CEO with high charisma with high levels of environment uncertainty resulted in a higher net profit margin, if the CEO had low charisma it resulted in a lower net profit margin. In times of low environment uncertainty the inverse relationship was found, high charisma resulted in lower net profit margin while low charisma resulted in a higher net profit margin. This shows how socio-human factors can be used as a tool to handle environment uncertainty.

3.4.6 Task Uncertainty

Task uncertainty are uncertainties related to ambiguous tasks or outcomes. High task uncertainty results in an unclear solution to the problem being solved (Dömmze and Grote, 2013). These uncertainties are common when working with new technology and solutions (Dömmze and Grote, 2018). Due to the rapidly changing global market task uncertainty will always exist and affect project teams (Rachel Dinur, 2011). The lack of knowledge surrounding a task might also result in a lack of responsibility. The flexibility which stems from agile methods also leads to uncertainty which reduces the feeling of ownership over a task (Conboy et al., 2012). The DAD framework proposes pair programming in order to increase knowledge sharing and getting a broader knowledge range (Ambler and Lines, 2012). By increasing team members knowledge and skills the quality of the solution increases and it also gives the team members the tools needed to handle unexpected difficulties (Dömmze and Grote, 2013).

During the process of turning requirements into tasks it is important to identify dependencies between the task in order to identify threats and opportunities (Dömmze and Grote, 2013). When lacking information about task dependencies it is possible to start with a task and then get stuck waiting for another task to be completed. The two option that exists are either to wait until you can continue or switch task. When starting different tasks without completing them, a switching costs is added, thus resulting in a longer implementation time (Cooper et al., 2015).
3.4.7 Technological Uncertainty

The technological uncertainty is based on whether projects use new and innovative technologies or mature technologies (Marinho et al., 2014). Innovative projects might base their infrastructure on technology which has not been created yet, thus creating a huge uncertainty. Such new technology might be hard to use, being a threat or the technology might be able to create solution parts which can be reused for other projects, thus being an opportunity. By working with new technology a team can expand their knowledge, and keep up to date with the latest technological trends.

Dvir and Shenhar (2007) identified four levels in which technological uncertainty can be divided into low-, medium-, high-, super high-tech. This scale is subjective and depends on the project team skills. Projects which are deemed as low tech have lower technological uncertainty and are less likely to fail, since the procedures are usually standardized (Dvir and Shenhar, 2007).

By using shorter iterations and following common agile practises technological uncertainties becomes more manageable (Marinho et al., 2014). The shorter sprints allow for new information being available more frequently and allow for more flexibility. Even though a distinction exists between uncertainties and risk, projects with a high technological uncertainty would benefit from risk management practices (Dvir et al., 2002). If technological risks are left unnoticed it can turn into uncertainties for the project, especially for super high tech ones (Marinho et al., 2014).

3.5 Partially Distributed Teams

Distributed teams or virtual teams are teams which are geographically dispersed and communicate through other means than face to face. Partially distributed teams are teams with at least one co-located subgroup and remote team members. Due to the geographic distance between all the team members, face to face meeting is not a viable option for all planning and communication. Whenever face to face meetings are opted for, one needs to take the cost, both time and money, into consideration (Conchúir et al., 2005).

Just like any IT project, partially distributed teams are subject to uncertainties. However, due to the nature of such teams uncertainties can arise from sources not even present for co-located teams. When comparing distributed teams with co-located ones, it has been shown that they experience more conflicts, especially task conflicts (Hinds and Mortensen, 2005). Conflicts may stem from threats or missed opportunities, thus highlighting the importance of being able to handle uncertainties.

3.5.1 Tools & Communication

One of the biggest issues with partially distributed teams is communication (Hiltz et al., 2013). The lack of communication can in turn lead to socio-human uncertainties. Common tools for partially distributed teams are directed to help with communication, since this is of great importance for any team (Conchúir et al., 2005). The lack of trust in infrastructure for communication plays a vital part in the communication within the team (Webster and Wong, 2008). If the communication tool is perceived as insufficient then the communication suffers.
One of the main goals for common infrastructural tools is to increase awareness by focusing on communication (Rodríguez et al., 2010). Without communication it can be easy to lose track of what other team members are working on. If there is a lack of awareness of what others are doing it might create bottlenecks based on dependency between tasks, stemming from task uncertainty. This is verified by Hiltz et al. (2013), who found that dependant tasks are even harder to handle with a partially distributed team. This might be the result of decisions being made in informal settings by the co-located group which does not reach the remote members (Webster and Wong, 2008). When taking breaks during works co-located team members still have the possibility to discuss the project and at times it can be hard to remember all changes which have been made in order to notify the remote members.

Even if face to face meetings are most effective video conferencing is a suitable substitute. Video conferences are more effective than calling, instant messaging and emailing (Hiltz et al., 2013). However due to poor video quality and internet connection this might not be a viable option for all partially distributed teams. Even organization with the needed equipment might have deemed the setup time as too costly for certain meetings. Webster and Wong (2008) also found that the local group of partially distributed teams communicate face to face as much as a co-located team. This in itself can lead to a reduction in information being passed to the remote group, thus creating socio-human uncertainties or task uncertainties if information regarding task are being communicated face to face. It has also been found that the cost of initiating contact through instant messaging is considered as greater than face to face, this results in choosing not to message team members (Hiltz et al., 2013; Conchuir et al., 2005). Even if the information is sent through a direct message to all team members it can not be guaranteed that the information is spread in a timely manner. It is much easier to ignore direct messages than face to face communication, thus people might not even read all information being sent to them (Hiltz et al., 2013).

When working at different geographical locations the informal communication reduces, resulting in a negative effect on the relationships between co workers (Conchuir et al., 2005). An informal communication can lead to a more natural flow of the smaller topics regarding a project and the result of this should not be diminished.

### 3.5.2 Group dynamic

Partially distributed teams, like any other type of team, can be either ongoing teams or temporary teams. Ongoing teams are teams with clear processes working with tasks over several years (Ahuja and Saunders, 2006). Ongoing distributed teams allow a learning and introduction process to take place. By giving the team members time to evolve and develop the processes and roles needed to complete the project, thus resulting in an increase of trust and giving the members time to solve conflicts (Kaufmann and Carmi, 2017). Temporary teams are teams working on temporary projects, usually high-skilled ones, working swiftly focusing more on the goal than leadership, management and team dynamics (Ahuja and Saunders, 2006). Temporary teams often have a short deadline and they are very task oriented, resulting in them being very efficient (Chae et al., 2015).

Since partially distributed teams have one co-located team with remote members there exists a difference with distributed teams. When having one co-located team with remote team members
an "Us vs Them" feeling exists (Hiltz et al., 2013). Such a feeling can easily result in distrust and competitiveness rather than cooperation. However, the feeling within the team differs depending on whether you are a part of the co-located team or are a remote team member. As a part of the co-located team a team member feels a sense of group identification at a higher level than the remote team members (Webster and Wong, 2008). However, a higher group dynamic results in lower perceived effectiveness for the team (Hiltz et al., 2016).

A negative effect of not having all team members at one location is that not all might share the same vision and goals for the project (Conchuir et al., 2005). When lacking common goal the result may be a higher degree of task uncertainty. The goals and visions for a project are always shifting, especially for agile projects, where team focuses on flexibility. When comparing co-located team members to remote members it has been shown that there exists a lack of natural communication (Conchuir et al., 2005). Natural communication is all communication taking place outside official projects meetings, this includes small talk while working and during breaks as well as time spent outside work. The lack of natural communication results in decreased trust among the team members (Conchuir et al., 2005). The geographical distance might make it hard to know what is happening at other locations. Not knowing what is happening may result in a lack of team awareness, which is common for partially distributed teams (Conchuir et al., 2005). However the nature of agile methods including daily stand-ups as a way to work against this. Daily stand-ups forces teams to communicate at least once a day and if proper tools such as a kanban board is used it is possible to see the actual results of other members (Webster and Wong, 2008).

Members of the co-located group tend to view each other more positively than the remote members (Webster and Wong, 2008). When working at different locations if might be hard to see what the other members are working on, thus increasing trust issues. Other sources of trust issues are those based on socio-human interactions (Webster and Wong, 2008). These trust issues might stem from actions such as muting microphones during calls and video conferences or not answering emails and instant messages. When not being able to see the other all team members, one might become more doubtful of the others skills (Conchuir et al., 2005). For teams to be able to trust each other one needs to be able to base trust on other factors than seeing them in the office between nine and five. It is also possible for the remote members to be part of other team constellations without the knowledge of members of the co-located group (Webster and Wong, 2008).

3.5.3 Management & Practices

Usually partially distributed teams are seen as the least preferred team constellation and if possible one should opt for either a distributed team or a fully co-located team (Webster and Wong, 2008). However, through effective leadership one can mitigate the negative effects of the team constellation while exploiting opportunities arising. When using members from different offices there might be a lack of concurrent engineering principles resulting in conflict. Hence in some cases it might be more beneficial to opt for standard practices rather than the self organizing teams which agile methods opt for. Distributed and partially distributed teams requires a standardization in work practices to ease early phases of the project and on-boarding processes (Conchuir et al., 2005).

An experienced leader can help create a group identification early in the project (Webster and Wong, 2008). Easy mean of communications and kick offs may increase the sense of belonging. When meeting team members in a more relaxed environment an increase of natural communication
occurs, thus resulting in increased trust among the team members (Conchuir et al., 2005). By having access to a larger labour pool it is possible to handpick a team based on skill and personal treats to make an effective team (Conchuir et al., 2005). However, if the knowledge management and information sharing is poorly implemented the advantages of having partially distributed teams are lost (Herbsleb and Moitra, 2001).

When working with different sub teams it is of great importance to divide the work efficiently between the different teams. This results in the creation of separate modules which can be used as separate components (Conchuir et al., 2005). These components can be used for other projects, thus resulting in a more efficient organization where parts of a solution can be reused for other projects. Another opportunity which exists for partially distributed team is that it might be easier to contact customers at different locations (Conchuir et al., 2005). If a remote team member works at the same geographical location as the customer it is easier to communicate with the customer hence exploiting requirement opportunities.

In order to decrease the in-group dynamic preparations can be made and training teaches the members how to interact within a partially distributed teams (Hiltz et al. 2016). Hence, it might be beneficial for a team to have more experienced with working in a partially distributed team to increase effectiveness as well as teach the new comers how to interact in such a team.

In a partially distributed team it can be hard to implement agile practices. However, agile principles can be beneficial for a project and result in a reduction of the sensed geographical distance (Conchuir et al., 2006). To ease the implementation of agile methods within distributed teams trust within the team is of great importance (Dorairaj and Noble, 2013). In order to build trust within a distributed team focus should lie on the team rather than the individuals in it, feedback needs to be given, regular communication needs to exist, skills and expertise needs to be demonstrated and members should take initiative to co-operate (Dorairaj and Noble, 2013). The techniques mentioned comes more naturally to a co-located, since it is easier to establish a relationship, hence it is of great importance for a leader of distributed team to implement the techniques.

In order to ease the implementation of agile practices an "ambassador" role should be added to the team, during the first iterations (Rizvi et al., 2015). The ambassador is a member of the co-located team who visits the remote members in order to ensure that the procedures are followed as well as to make sure that the on boarding process does not differ between sites (Turcek et al., 2010). The co-locations of teams have shown to be one of the best ways to mitigate risks associated with distributed agile teams as it increases communication, trust and understanding among team members (Rizvi et al., 2015).
4 Methodology

In this chapter the methodology is presented, explained and motivated. An overall description is first given. This is followed by a detailed description of the data gathering and analysis. The chapter ends with a motivation of how reliability, validity and generalizability is achieved.

From the research purpose and questions it can be concluded that the study is going to be a qualitative study within the interpretivism paradigm. The interpretivism paradigm exists under the belief that the findings are biased and that knowledge comes from subjective evidence of participants (Collis 2014). In order to answer the research questions one needs to study how agile methods, partially distributed teams and uncertainties affect one another. Hence, a complex relation between several factors is to be studied, thus making the case study method a natural choice (Denscombe 2010). To ensure the ethics of the research, it has been designed according to the guiding principles by Bell and Bryman (2007), see section 1.6.

4.1 Research Design

The purpose of this study is to research current uncertainties at a company, a real life context, which fits well with the scope of a case study (Yin 2009). To make the research more robust multiple case studies will be performed on two cases with similar traits. In order to build a theory from a case study, it needs to be possible to replicate it and the more replications that are done the stronger the theory is (Eisenhardt and Graebner 2007). There exists five main stages of a case study (Collis 2014):

1. Selecting the cases: The case company was selected in October 2017 and was chosen based on access. Two cases were chosen for the research. Both of these fulfilled the following three criteria:
   (a) The teams should be partially distributed
   (b) The teams should work on in-house projects
   (c) The teams should consist of at least three people
2. Preliminary investigations:
   (a) A literature study was conducted in order to gain an understanding of the background for the questions being studied. See section 4.2
   (b) A pilot study was conducted in an early stage to understand the company culture and the structure of in-house projects. See section 4.3.2
3. Data collection. See section 4.3.3
4. Data analysis. See section 4.4
5. Writing the report. This step will be done in parallel to all the steps above. The process of writing will be an iterative process according to the guidelines from Blomkvist and Hallin (2015).

The different cases will first be analyzed separately with individual results and analysis and after that it will be possible to draw conclusions based on the two cases (Yin 2009).
4.2 Literature Review

The literature search was based on the methods by [Watson and Xiao (2017)](https://pmt-eu.hosted.exlibrisgroup.com/primo-explore/search?vid=46KTH_VU1_L&sortby=rank&lang=en-US). First keywords were identified and searched for. Related keywords to subjects which were searched for were: "IT project uncertainties", "poor agile management", "agile software development", "implementing agile", "software risk management", "distributed teams", "distributed agile", "partially distributed teams", "distributed team", "software project uncertainties", "lean software development", "resource uncertainty", "requirement uncertainty", "task uncertainty", "technological uncertainty", "environmental uncertainty", "market uncertainty" and "socio-human uncertainty". These terms were searched using KTHB Primo<sup>2</sup> and Google Scholar<sup>3</sup>.

The articles found by using the keywords listed above were then screened for inclusion. By looking at the title first and then the abstract, it is possible to rate the level of relevance of an article found (Watson and Xiao, 2017). At the final stage the entire article was read and if still deemed as relevant it was shortly summarized in a document. If an article was deemed relevant both a backward search and forward search was conducted (Watson and Webster, 2002). In order to do a backward and forward search the articles references citings functions were used.

All read literature passing the screening of the abstract were added to an excel sheet, with the URL and title of article. This was to keep track of all articles, so that none were read more than once and to establish a structure over the literature study. The literature review was conducted in order to gain a deeper understanding of the subject and find different areas of uncertainty.

4.3 Data Gathering

4.3.1 Interviews

The main method for data gathering for both the pilot study and the multiple case studies was interviews. Before all the interviews a protocol consisting of the interview structure was created. It consisted of three parts: description of the research, establishing consent and questions (Arsep, 2017). In order avoid bias or influencing the interviewee it is important to not state your hypothesis. The questions themselves were created in an iterative process, where a person not related to the company

---


<sup>3</sup>https://scholar.google.se

Figure 2: Gant Schedule of Master Thesis period
in any way was interviewed. This was to establish that the questions were clearly formulated (Arsel, 2017).

The interviews were all one-to-one interviews. This was based on four factors (Denscombe, 2010):
1. It is easier to schedule an interview the fewer people who are participating
2. The interviews become easier to transcribe
3. Each person get equal opportunity to express their thoughts without the pressure of someone else listening
4. It becomes easier to ask follow up questions.

When conducting interviews it is important to gain trust of the interviewee (Handfield et al., 2002). The trust gained among the interviewees differed for this research, this was due to the fact that most observations were made at the Stockholm office. The time spent at the Stockholm office was at times a bit informal, with activities such as eating lunch and talking informally with the subjects, thus resulting in a greater trust.

4.3.2 Pilot Study

The pilot study was used to narrow the purpose of the study as well as give a greater understanding of the organization to be studied. After the previous step it was also possible to start with the more exhaustive literature review. Main method used for this were interviews. However, time was also spent at the Stockholm office of Dewire and used observations as part of gathering knowledge of the organization. This was to get a clearer view of the context which is to be studied.

Introductory Interview

An introductory interview was conducted in order to figure out which agile method the organization had implemented. This interview was used for gaining an understanding of the problem and getting to learn the basics of the company. The interview was transcribed, but not recorded. The interview was semi-structured, leaning a bit more to the structured part of the spectrum (Denscombe, 2010). The interviewee was chosen based on availability and had worked at Dewire for roughly two years.

<table>
<thead>
<tr>
<th>Role</th>
<th>Location</th>
<th>Date</th>
<th>Duration</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tester</td>
<td>Stockholm</td>
<td>21/12-17</td>
<td>30 min</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Overview of conducted introductory interview
Navigational Interviews

Navigational interviews were conducted in order to understand team dynamics, work procedures and how the company functions. See Appendix B for interview questions. The interviews were semi-structured meaning that the questions were used as a guide and may not have been answered in the presented order (Denscombe, 2010). Before the end of the interview a quick overview was completed in order to ensure that all relevant topics had been covered. The interviewees were between the ages 26 and 47 and they had between two and 14 years of experience of working at Dewire.

<table>
<thead>
<tr>
<th>Role</th>
<th>Location</th>
<th>Date</th>
<th>Duration</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>Stockholm</td>
<td>5/02-18</td>
<td>20 min</td>
<td>1</td>
</tr>
<tr>
<td>Developer</td>
<td>Sundsvall</td>
<td>13/02-18</td>
<td>21 min</td>
<td>2</td>
</tr>
<tr>
<td>Solution architect</td>
<td>Sundsvall</td>
<td>13/02-18</td>
<td>17 min</td>
<td>3</td>
</tr>
<tr>
<td>Project manager &amp; Scrum master</td>
<td>Sundsvall</td>
<td>13/02-18</td>
<td>34 min</td>
<td>4</td>
</tr>
<tr>
<td>Project Manager</td>
<td>Sundsvall</td>
<td>13/02-18</td>
<td>23 min</td>
<td>5</td>
</tr>
<tr>
<td>Developer</td>
<td>Stockholm</td>
<td>15/02-18</td>
<td>33 min</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 2: Overview of conducted navigational interviews

4.3.3 Multiple Case Studies

As mentioned the research consisted of two case studies. However, due to the limited time these were done in parallel with a single pilot study for both of them. Using multiple sources for data gathering allows for triangulation, but it also increases the resources needed for a research (Yin, 2009). Due to this one main source was chosen. In order to avoid the site bias, interviews were chosen as the main method. This also gives the possibility to study two projects rather than one.

Exploratory Interviews

These interviews were semi-structured, leaning a bit more to the unstructured part of the spectrum (Denscombe, 2010). Topics to be discussed were noted before the interviews and back-up questions in order to get the interviewee to talk existed and can be found in Appendix F. The aim with these interviews was to discover which uncertainties exist and how they were being managed at the moment, thus focusing on the interviewee thoughts (Denscombe, 2010). The topics to be discussed were: general project description, the interviewee’s role in the project, communication within the team, communication with other stakeholder, management practices and uncertainties related to the project.

These interviews were exploratory in order to find the different uncertainties and how they are currently being handled. In order to avoid biased answers multiple project members were interviewed and two different cases were used. The goal with these interviews is to find uncertainties and how they are handling them at the moment. Between five and seven days before the interviews in Sundsvall, the interviewees were sent a short informational mail regarding the topic of the master thesis and the topic for the interview. For the interviewees in Stockholm the basic topic of the interview and the purpose of the research was shortly explained roughly one week before the interviews. To be noted is that interviewee H and interviewee I had their interviews postponed due to personal reasons, hence they were noted of the interview topics between two and four weeks before their interviews.
In order to get a representative view of the project a mix between the roles, at least one team member had to be located in Stockholm and one in Sundsvall. From all projects a project manager, a lead (either test, developer or both), scrum master, developer and tester were interviewed.

Since Project 1 is lacking a scrum master at the moment, one could not be interviewed. Due to the fact that the project is divided into two different sub-teams, at least one member from each team was interviewed. The interviewees were between the ages 25 and 51 and they had between one and 18 years of experience of working at Dewire.

<table>
<thead>
<tr>
<th>Role</th>
<th>Location</th>
<th>Date</th>
<th>Duration</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architect</td>
<td>Sundsvall</td>
<td>27/03-18</td>
<td>52 min</td>
<td>A</td>
</tr>
<tr>
<td>Project Manager</td>
<td>Sundsvall</td>
<td>27/03-18</td>
<td>52 min</td>
<td>B</td>
</tr>
<tr>
<td>Team lead (front end)</td>
<td>Sundsvall</td>
<td>27/03-18</td>
<td>55 min</td>
<td>C</td>
</tr>
<tr>
<td>Developer (back end)</td>
<td>Sundsvall</td>
<td>27/03-18</td>
<td>48 min</td>
<td>D</td>
</tr>
<tr>
<td>Test lead</td>
<td>Sundsvall</td>
<td>28/03-18</td>
<td>48 min</td>
<td>E</td>
</tr>
<tr>
<td>Developer (front end)</td>
<td>Sundsvall</td>
<td>28/03-18</td>
<td>55 min</td>
<td>F</td>
</tr>
<tr>
<td>Tester</td>
<td>Stockholm</td>
<td>16/04-18</td>
<td>55 min</td>
<td>G</td>
</tr>
</tbody>
</table>

Table 3: Overview of conducted interview for Project 1

The interviewees were between the ages 27 and 36 and they had between two and five years of experience of working at Dewire.

<table>
<thead>
<tr>
<th>Role</th>
<th>Location</th>
<th>Date</th>
<th>Duration</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project manager &amp; Scrum master</td>
<td>Sundsvall</td>
<td>02/05-18</td>
<td>53 min</td>
<td>H</td>
</tr>
<tr>
<td>Developer</td>
<td>Sundsvall</td>
<td>28/03-18</td>
<td>40 min</td>
<td>I</td>
</tr>
<tr>
<td>Architect</td>
<td>Sundsvall</td>
<td>28/03-18</td>
<td>53 min</td>
<td>J</td>
</tr>
<tr>
<td>Tester</td>
<td>Stockholm</td>
<td>16/04-18</td>
<td>55 min</td>
<td>G</td>
</tr>
<tr>
<td>Team lead</td>
<td>Stockholm</td>
<td>16/04-18</td>
<td>45 min</td>
<td>K</td>
</tr>
<tr>
<td>Developer</td>
<td>Stockholm</td>
<td>12/04-18</td>
<td>62 min</td>
<td>L</td>
</tr>
</tbody>
</table>

Table 4: Overview of conducted interview for Project 2

Observations
By spending time at the Dewire office in Stockholm observations were made. Field trips to the Sundsvall office were made in order to conduct interviews and observe the environment. The observations were a mix of participant and direct observation. When observing meetings direct observation was used, this was to see the reality of the meetings without them being affected by the researcher. Other than during meetings participant observation was used, this was to gain more insight into behavior and motives (Yin, 2009). However, these observations were mostly centered around the Stockholm office making them biased, which is something that needs to be taken into account.
4.4 Data Analysis

The analysis of the interviews from the pilot studies and the case study followed the same structure (Collis 2014).

- Reducing the data
- Restructuring the data
- Contextualizing the data

After these three steps it is possible to display the data in order to draw conclusions from it (Collis 2014).

4.4.1 Pilot Study

As a starting point a comparison between Dewire’s project management method and traditional DAD will be made. Due to the nature of the purpose of the pilot study the data could be reduced and reconstructed to a matrix using an excel sheet. The matrix could then be used to make comparison and contextualization simple. The analytic process of developing a case description is used to present the result. This is due to the fact that no clear questions were to be answered during the pilot study, making it hard to rely on existing frameworks (Yin 2009). However, a thematic analysis based on the responses will be used to display the data (Blomkvist and Hallin 2015).

The results were first analyzed as individual responses. These responses were then compared by home office and roles. This was to see if there were any standard practices as well as to identify any discourse between the employees.

4.4.2 Multiple Case Studies

The purpose of the case studies is to find which uncertainties partially distributed teams are subject to and how they manage them. In order to first identify which uncertainties they are subject to a thematic analysis, based on the literature study, will be used (Blomkvist and Hallin 2015). Color codes will be used to restructure the data and getting an overview. The codes to be used will be based on the uncertainties found in section 3.4. See Appendix D for the codes used for data analysis.

Research questions 1 and 2 will be answered through an explanation building strategy. This is well suited to this case study as no hypothesis has been established (Yin 2009). The uncertainties found will be compared to those in section 3.4 using a pattern matching logic. Pattern matching compares the empirical data gathered with alternative predictions in order to find what type of uncertainty an unplanned event can be classified as (Yin 2009). Once uncertainties have been identified a comparison between how agile and lean methods which are used to increase flexibility will be compared with how the teams managed the uncertainties. The main objective of the observations is through triangulation ensure that the truth is said by the interviewees. Quotes
will be presented in order for the reader to understand the interpretations. However, it should be noted that the quotes have been translated from Swedish to English.

A method which is to be used is frequency. The uncertainties within each project will be ranked according to frequency, this will allow the comparison between different roles and office location. In addition, this will make it possible to focus on the most common uncertainties which the whole team is affected by.

Once the two cases have been analyzed separately, the results of those can be synthesized to draw common conclusions. The results of doing a cross case analysis is that the analysis becomes simpler and more robust (Yin, 2009). The comparison can also be used to reduce the uncertainties which are project specific and can help find those which are more general for the context (Eisenhardt and Graebner, 2007). In order to compare the two cases a word table will be used in order to display the identified uncertainties from the two separate cases (Yin, 2009). The results found will be discussed with a representative from the company in order to identify any rivalry explanation and interpretation.

4.5 Validity & Reliability

The quality of a research is established through construct validity, internal validity, external validity and reliability (Yin, 2009). This is to ensure that the correct phenomena is being studied, that it is being studied in the correct way and that the analysis has been done with high quality (Blomkvist and Hallin, 2015).

Construct Validity
Construct validity is based on how data is gathered and if key informants review the analysis draft (Yin, 2009). After the pilot study the result will be shown to an employee at Dewire, and any issues which are brought up will be documented and added to the report. This is to verify that the data collected does represent the organization. Construct validity is also strengthened in the thesis by using multiple data collection methods, such as interviews and observations (Frohlich et al., 2002).

Internal Validity
Internal validity is how data is analyzed, pattern matching, explanation building, the use of logical models and addressing rival explanations (Yin, 2009). During a case study it might be hard to find what causes a specific event to happen, but by following certain analysis methods the internal validity is strengthened. Before finding a causal relationship, all rival explanations needs to be considered.

External Validity
External validity is how replication logic is used on the multiple-case studies, which is established in the research design and how the results can be generalized (Yin, 2009). Still, it is clear that a case study by itself can never result in statistic generalizibility. However, it is possible to get analytic generalizability (Blomkvist and Hallin, 2015). The analytical generalizability can then be used as a base for either a hypothesis or a theoretical proposition (Yin, 2009). To allow contextualization, the context is described in section 5.1.
Reliability

Reliability is important as it makes it possible for other researchers to replicate the study (Blomkvist and Hallin, 2015). In order to establish reliability a case study protocol recommended by Yin (2009) can be found in Appendix C. Yin (2009) recommends using a case study database, due to nature of the signed non disclosure agreement this will not be possible within the limited time of the research. If a database was to be established all notes would have to be censured and then be read through first by the interviewees and then by a company representative.
5 Empirical Result

This chapter will bring a summary of the empirical results. It starts with a description of the context and a description of the two cases being studied. This is followed by the result of the pre-study and case studies. The data is presented in the same order it was gathered due to the complexity of the research.

5.1 Context

The case study was performed at Dewire, a medium size IT-consultancy firm founded in 1997. The 76 employees are spread geographically, with 65 people located in Sundsvall and 11 people in Stockholm. Due to this, several projects which they take on are completed by a partially distributed team. From the interviewee it was concluded that the agile method used was DAD. The projects that Dewire takes on are both development project and administration projects. Recently Dewire started with the development of their own service, to which subscriptions will be sold. The team size and constellation usually vary between different projects.

Sundsvall office: The Sundsvall office have all employees in different rooms, which room an employee belongs to depends on their current project. In practice this results in an open office landscape within each project. Several meeting rooms exist, but due to the fact that they have more employees these needs to be booked.

Stockholm office: The Stockholm office has an open office landscape where all employees sit. They have a few conference room where they hold interviews and have meetings in, this makes it possible for the employees to work in a temporary office.

Communication for Dewire is handled through the channel based application slack. The communication in slack exists in different channels, allowing transparency if wanted. In order to see the projects status a virtual kanban board is used, namely JIRA. By using JIRA, comments can be added to each task specifying errors or new requests.

In order to understand the result of the case study the context of the two projects being studied needs to be given. Both projects consists of a co-located subgroups in Sundsvall with remote members in Stockholm.

Some of the terms used by Dewire differ from traditional agile terms, hence they will be explained here:

- **Epic**: A term which Dewire uses to describe the requirement formulated by the customer.
- **Story**: A term which Dewire uses instead of a task. A story can be divided into sub-tasks.
**Project 1**

Project 1 is one of the larger projects at Dewire. The project is also described as a program due to the length of. Dewire has been working with the project during the last six years. Due to this they have had a rotation of project members and very few have worked with the project from start to finish. The project is divided into two teams, front- and back- end, with their own testers, test leads, developers and lead developers. Other roles within the project are: project manager, solution architect and scrum master. However, for the past few months the project has been lacking a scrum master. Tools used to aid the development for this project is also slack and JIRA. The management style is DAD with the goals of having daily stand-ups, two times a week with both sub-teams and three times within the sub-teams. The project is supposed to use 8.75 resources, according to the customer.

**Project 2**

At the moment the project is very small. Dewire is functioning as a support function fixing bugs and no new features are being added at the moment. The project started in 2015 and is within the telecom industry. The project has been based from Stockholm with people joining from Sundsvall from time to time, thus retrospectives were usually held in Stockholm. Tools used to aid the development for this project is slack and JIRA, but when new functionality was added the project used Skype as the main communication tool. The management style for this project has changed, starting with DAD, but due to changes to the project it has become an ad hoc management style.

**5.2 Pilot Study**

The pilot study results are based on the results from interviewees 1-6 and observations done between the weeks 4 and 8. During the interview three main topics were identified, these are: project management, technological issues and documentation & communication.

**5.2.1 Project Management**

Interviewees 1-5 all agreed on that the method used was DAD, interviewee 6 did not know which method they used but mentioned that it was agile. Both interviewees 4 and 6 thought the agile method used in practice was a bit ad hoc and interviewee 5 mentioned that they were flexible with their method choice. Interviewee 3 made it clear that in theory they use DAD, but not in practice. Interviewee 4 believed that the reason they want to use DAD is because it has an initialization phase.

All interviewees mentioned that they had daily stand-ups and interviewees 1-5 all agreed on that they spent 10-20 minutes for these. However, interviewee 6 mentioned that they were between 20 and 60 minutes most of the times. They all thought that these were beneficial. Projects with sub teams had two different types of daily stand-up alternating every other day, one with their respective sub team and one with all the sub teams. However, through observations it has been noted that at least three stand up were canceled due to unknown reasons.
All interviewees agreed that they had retrospectives, but when asked how often the answer varied. Interviewees 2 and 6 said that they did not have retrospectives after every sprint, while the rest of the interviewees stated that they had them after every sprint. All interviewees agreed on that these were beneficial and that these brought up relevant issues. Interviewees 1, 5 and 6 mentioned that the retrospectives have been improved due to a single individual they praised this individual for the quality of the retrospectives. According to interviewees 4 and 3 the teams also uses futurespectives for sprint planning. Interviewee 4 explained that these are usually planned for when the Stockholm employees travel to Sundsvall.

5.2.2 Technological Issues

The response from the interviewees differed in regard to technological issues, but all agreed that technological issues might cause meetings to start late. Interviewee 1 said that the daily sprints started between 10 and 15 minutes late due to technological issues, but that the situation has been improved. Interviewee 2 mentioned that the daily sprints starts a bit late due to technological issues and that roughly once a week meetings are canceled because of this. Interviewee 3 mentioned that meeting sometimes may start late due to technology, but this mostly has to do with setup time, rather than technological issues. Interviewee 4 said that it has been a learning process, after a while they have learned which microphones works with which computer and therefore reduced the overall setup time, but meetings might start late due to technological issues. Interviewee 5 said that meetings usually start on time, this was also the first interviewee to mention that they have a video conference corner, a station where everything is already set up. This corner has reduced setup time and now meetings are only affected by technological issues once every other week. Interviewee 6 mentioned that the main technological issue during meetings is sound not working, which is both a slack problem and problem with the microphones.

5.2.3 Documentation & Communication

Interviewees 1, 4 and 6 all agreed on that the main communication method is slack. However, interviewees 2 and 5 believed that face to face communication was mostly used, followed by slack. By having slack they create clear channels for communication, with forums for locations, projects and less serious channels. Interviewee 1 was the only person to bring up the negative sides of using slack; it is hard to find old messages sent on slack and external users can see everything. Interviewee 4 stated that the use of slack is pretty new and previously they had used Skype. Interviewee 6 mentioned that private messages sometimes are sent on slack, when not everyone needs to be notified of something and you need to think carefully who needs to see the message before you send it.

During the pilot study it has been observed that the employees at the Stockholm office talks in terms of ”Us vs Them”. A few examples being ”it’s their responsibility”, ”they’re eating lunch in Sundsvall now, so I can’t do anything”. When asking if there existed an ”Us vs Them” feeling, interviewee 0 verified it.

In regards of documentation all interviewees had different views on what was documented and how it was documented. According to interviewee 6 an official rule stating that slack does not count as documentation exists, but this is not followed. This was also verified by interviewee 1 who said that
it has been decided that all decisions needs to be made official on mail, slack and oral decision were not valid. Both interviewee 2 and 3 document decisions in JIRA, however these decisions can be made orally or over telemeetings. Interviewee 5 stated that most decisions were made orally, but did not know if and how it was documented.

Through observations it has been noted that not all daily stand ups are documented. Both interviewee 2 and 4 verified that retrospectives for their respective projects were documented by taking pictures of Post-it notes at the end. These photos were then uploaded, making it possible for all team members to look at them. However, interviewee 2 mentioned that he/she had never looked at those pictures. According to interviewee 4 the reasons for not taking notes during the retrospective is because it hinders natural conversations and makes it hard to ask follow up questions.

Interviewee 6 brought up a single case where documentation was used to their advantage. During this project a journal was kept, in this journal all decisions were documented both between the team members and between the team and the customer. The customer were displeased with a function in the product and blamed Dewire, but when going through the journal it was possible to see that the function was requested by the customer.
5.3 Project 1

In Table 5, the uncertainties or sources of uncertainties identified during the interviews are presented. These uncertainties are based on the interviews as well as observations. The states and outcomes mentioned in this table are those which were unplanned.

<table>
<thead>
<tr>
<th>No.</th>
<th>Uncertainty/ Source of uncertainties</th>
<th>Interviewees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Not constant communication with Stockholm</td>
<td>A, B, C, D, E, F, G</td>
</tr>
<tr>
<td>2</td>
<td>Ad hoc agile</td>
<td>A, B, C, D, E, F, G</td>
</tr>
<tr>
<td>3</td>
<td>Customers are not agile</td>
<td>A, B, C, D, E, F, G</td>
</tr>
<tr>
<td>4</td>
<td>Team members are specialists</td>
<td>A, B, C, D, E, F</td>
</tr>
<tr>
<td>5</td>
<td>Requirement quality varies</td>
<td>A, C, E, F, G</td>
</tr>
<tr>
<td>6</td>
<td>No documentation guidelines</td>
<td>A, C, E, D, G</td>
</tr>
<tr>
<td>7</td>
<td>Unclear/ambiguous tasks</td>
<td>A, B, C, D, G</td>
</tr>
<tr>
<td>8</td>
<td>External systems have bugs and not sufficient documentation</td>
<td>A, E, F, G</td>
</tr>
<tr>
<td>9</td>
<td>Task dependencies</td>
<td>A, C, D, F</td>
</tr>
<tr>
<td>10</td>
<td>Internal customer requirement ambiguity</td>
<td>A, C, E, F</td>
</tr>
<tr>
<td>11</td>
<td>Slack messages are sometimes ignored</td>
<td>A, C, D, E</td>
</tr>
<tr>
<td>12</td>
<td>Hard to estimate tasks</td>
<td>B, C, E, F</td>
</tr>
<tr>
<td>13</td>
<td>Unclear how to handle mid sprint requirements</td>
<td>A, B, C, F</td>
</tr>
<tr>
<td>14</td>
<td>Remote members lack information</td>
<td>A, D, E, G</td>
</tr>
<tr>
<td>15</td>
<td>Customer do not have sufficient resources</td>
<td>D, E, G</td>
</tr>
<tr>
<td>16</td>
<td>Testers communicate with developers if uncertain</td>
<td>C, E, G</td>
</tr>
<tr>
<td>17</td>
<td>GDPR</td>
<td>A, E</td>
</tr>
<tr>
<td>18</td>
<td>Test and production environment are different</td>
<td>G</td>
</tr>
<tr>
<td>19</td>
<td>More resources needed than customers wants</td>
<td>B</td>
</tr>
<tr>
<td>20</td>
<td>Had to redo Apple Watch App</td>
<td>F</td>
</tr>
</tbody>
</table>

Table 5: Overview of uncertainties and sources of uncertainties

5.3.1 Communication, Decisions & Documentation

Interviewee E stated that "80% of the communication is probably oral..." and all participants agreed that the remote members in Stockholm are not a part of all discussions and conversations. Not constant communication with Stockholm is in itself not uncertainty, it can either be the result of an uncertainty, or the source from which other uncertainties stem from. Interviewees A, D, E, G went on and explained that this usually resulted in the remote members lacking project specific information.

Interviewees A, C, D, E noted that sometimes slack messages are ignored, not out of malice, but because they are considered as a disruption. Interviewee C stated that "...direct communication can be a distraction..." which is why they sometimes do not answer slack messages directly.
Most of the decisions are usually made orally or through Slack according to all interviewees. However, according to interviewees A, C, D, E, G it is unclear what actually gets documented. Interviewee C said "...we don't document enough..." and interviewee F verified that "...all decisions regarding stories are not added to JIRA, some things fall through the crack..."

5.3.2 Agility

Ad hoc agile is the summary of statements which do not correspond with traditional agile methods. Even though all participants agreed that they had daily stand-ups it has been observed at five separate occasions that the Stockholm office have not been a part of these. Interviewees A, B, C, D, E, F, G all agreed on that they have retrospectives, but not after every sprint. It was unclear how often they had retrospectives.

Even though the project is considered as agile by all team members, the customer of the project do not have continuous releases. They have between two and three releases each year. In practice this results in a bottle neck right before the solution is released. Functionalities might not be tested until six months after they are completed and then they are sent back to Dewire for correction. Interviewee A stated that "...the biggest issue with the project is that the receiving organization is not agile, but have processes from another era, with waterfall processes...".

5.3.3 Requirements, Resources & Tasks

When the team only receive one-liner requirements then it is unclear what is expected and what they are supposed to produce. Interviewee F stated that "...the received requirements are very bad, we almost have to write the requirement ourselves...". Interviewee E stated "...the requirements are not the most clear ones...". However, during the past year the quality of the requirements have improved, interviewee C said "...the last three or four months we have gotten clearer epics and therefore clearer stories...".

When asked how mid sprints requirements were handled, the answer differed and no clear explanation was given. Interviewee A stated that if something is changed mid sprint "You pull the Toyota string" and change the course of direction. Interviewee B stated that previously they implemented mid sprint requirements during the current sprint, this have changed now, today they don not accept changes to requirements mid sprint, except for small changes. Interviewee C stated that if it is big changes then they have to stop with what they are doing. Interviewee F stated that if it is a contradicting requirement or a completely new one they postpone it to the next sprint and the customer has to "...blame themselves...", smaller requirements can be added to current sprint, since their goal is to please the customer.

Internal customer ambiguity stems from that the requirements are discussed in Sundsvall, with the customer, but are then tested in Stockholm by the customer. This results in tasks being sent back to the developers and unplanned bug fixes.

Tasks are not always easy to estimate. They try to break down tasks so they do not take more than one day, preferably a few hours. They have noted that tasks are easier to estimate if they are smaller. Interviewee F brought up a situation in which they had underestimated a task "...we had
a big story which we analyzed and underestimated. We were really stressed at the end of the sprint and it took longer than expected...”.

Interviewee D, E, G all stated that Dewire’s own testers are sometimes leased out to the customer to aid with acceptance test. Interviewee G continued by saying “...it shouldn’t be like that, but they needed the resource then...”. However, interviewee D stated that this could be used as an advantage since Dewire’s tester could communicate directly with the developers if a bug is found.

Interviewee B stated that the project is supposed to use 8.75 resource, but in order to complete all tasks it is currently using 10. The reason behind these is explained by interviewee B who states that they also take care of maintenance of the system, which adds resources.

5.3.4 Externalities

Two interviewees stated that their project have been slightly affected by GDPR. Both stated that the effect was small. Interviewee E stated that this would only affect the crash log of the apps, this is because when an app crashes data is sent to a server where it can be reviewed. However according to interviewee E the data stored from crash logs is deleted fairly often, so as long as they notify the customer it should not be too much trouble.

The first watchOS app was created for Apple watch watchOS 1. When the operating system upgraded to watchOS 2 it affected the project. According to interviewee F they had to redo the entire app due to the differences between the operating systems. Interviewee F stated “...we had an old apple watch app, the differences [between watchOS 1 and watchOS 2] were huge and we had to scrap the app...”.

The project uses another system for data according to interviewees A, E, F, G and this system does in some cases cause trouble. Interviewee A stated “...the project is dependant on external systems... and sometimes these system differs from how we interpreted them...”. “...we have to adjust to [external system]... ...but we can’t trust the documentation blindly... ...bugs found are reported to [the customer]... ...[the customer] has a longer lead time to [the external system], it would benefit everyone to be more flexible...” are all statements from interviewee E.
5.4 Project 2

In table 6 the uncertainties or sources of uncertainties identified during the interviews are presented. These uncertainties are based on the interviews as well as observations. The states and outcomes mentioned in this table are those which were unplanned.

<table>
<thead>
<tr>
<th>No.</th>
<th>Uncertainty/ Source of uncertainties</th>
<th>Interviewees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Modularisation of tasks</td>
<td>H, I, J, K, L</td>
</tr>
<tr>
<td>2</td>
<td>Key team members with information</td>
<td>G, H, I, J, L</td>
</tr>
<tr>
<td>3</td>
<td>Customer had long response time</td>
<td>H, I, K, L</td>
</tr>
<tr>
<td>4</td>
<td>Unclear/ambiguous tasks</td>
<td>G, K, L</td>
</tr>
<tr>
<td>5</td>
<td>Not always easy to find needed documentation</td>
<td>G, I, L</td>
</tr>
<tr>
<td>6</td>
<td>Not enough communication with the customer</td>
<td>G, H</td>
</tr>
<tr>
<td>7</td>
<td>Faulty guidelines for external system</td>
<td>G, L</td>
</tr>
<tr>
<td>8</td>
<td>Project lost resources for two weeks</td>
<td>H, L</td>
</tr>
<tr>
<td>9</td>
<td>Project suddenly stopped</td>
<td>G, L</td>
</tr>
<tr>
<td>10</td>
<td>Ignored direct messages</td>
<td>I, J</td>
</tr>
<tr>
<td>11</td>
<td>Poor communication within team</td>
<td>I, J</td>
</tr>
<tr>
<td>12</td>
<td>Unrealistic requirements</td>
<td>J, K</td>
</tr>
<tr>
<td>13</td>
<td>Stockholm missed a few daily standups</td>
<td>I, K</td>
</tr>
<tr>
<td>14</td>
<td>System worked differently in production environment</td>
<td>J</td>
</tr>
<tr>
<td>15</td>
<td>Unclear division of work between two project managers</td>
<td>H</td>
</tr>
<tr>
<td>16</td>
<td>Stockholm office continued working on project after it was completed</td>
<td>H</td>
</tr>
<tr>
<td>17</td>
<td>Last minute requirements</td>
<td>H</td>
</tr>
</tbody>
</table>

Table 6: Overview of uncertainties and sources of uncertainties

5.4.1 Communication, Decisions & Documentation

During the project documentation caused a bit of issues. Interviewees G, H, I, J and L stated that the project had one or two key members. These members had been working with it for awhile and knew how the solution was supposed to work. Interviewee I said "...if something was unclear I went and asked [a team member], who had been with from the start, how it was supposed to work...". This shows that the information existed, but it was implicit. However, Interviewees G, I and L stated that documentation existed, but it was hard to find what you needed.

The Stockholm members of the team felt that there were no issues with the communication. However, Interviewees I and J said that they felt that the communication within the team should have been better. Interviewee I said "...my communication with the Stockholm members was non existent...". This also showed itself by a discourse existing between the Sundsvall office and Stockholm office. Interviewee K stated that they had stand ups every other, while all other members said that they had daily stand ups. Interviewee I also said "...the stand ups where out of sync between the location, resulting in us having our own...". One of the bigger issues mentioned by Interviewee H was that
the Stockholm office continued working with the solution and implementing new requirements after the project had been handed over to the customer.

For a while this project had two project managers, this turned out to be quite problematic. Interviewee H had contact with the customer, but was aware that the other project manager also had contact with them, but no real consensus regarding responsibility existed. Interviewee H said ". . . I called the customer to clarify requirements, [the other project manager] also did that . . . ."

5.4.2 Requirements, Resources & Tasks

Interviewees H, I, J, K and L all stated that their existed a modularisation of tasks making it easier to work in a partially distributed team. The main modularisation was having division by back end and the different front end clients. Each module was co-located and Interviewee L stated that ". . . me and [another team member] worked with the apps and [two other team members] worked with the web client . . . ."

Interviewees G, K and L stated that the tasks were a bit unclear during the project. Interviewee K stated that unclear tasks stemmed from unclear requirements. Another task related issue mentioned was that when team members needed a clarification it was hard to get a hold of the customer causing members to start on a new task. Interviewee H said that "...in order to get answers from the customer I had to write: should we do A or B, if you do not answer we will do A . . . .", thus it was at times hard to get a clarification of ambiguous tasks stemming form ambiguous requirements.

The projects had a few issues with requirements. Interviewees J and K stated that the requirement were unrealistic or hard to implement. Interviewee J went said "...A common mistake is that the people who write the requirements aren’t programmers . . . . Another issue was that last minute requirements were added to the project at the end according to Interviewee H. The most urgent requirement added was documentation, the project manager was not aware of this until the end of the project. The customer also added new requirements after the project was considered as done resulting in discourse between the Sundsvall and Stockholm office.

The project had a low priority at Dewire due to lack of commitment of the customer. Interviewee H said ". . . [the contact person at the company] told me: 'sorry that I have not been easy to get a hold of, we have had lots to do'. [The customer] were busy with some new EU regulation . . . ." and "...because the customer were absent, both in communication and financing other projects were prioritized . . . .". The lack of commitment from the customer lead to Dewire prioritizing other projects, thus the projects lost all resources for two weeks. This lack of resources becomes apparent when key members, members with lots of knowledge, are moved to another project. Interviewee H said "...during the Monday production management meeting I had to fight for [two team members] . . . ."

5.4.3 Externalities

For this project the customer became a hurdle. Interviewees G and H stated that there was not enough communication with the customer and Interviewees H, I, K and L stated that the customer

---

6A meeting with all project managers to plan resources for each project
had a long response time. This also caused issue with the faulty guidelines for the external system. Interviewee L said "...we needed to be in contact with [the customer] to debug the [external system]..." this became difficult due to the difficulty of contacting the customer. The lack of communication also made it hard to get clarification of requirements.

Another uncertainty based on external factors was that the production environment worked differently from the testing environment according to Interviewee J. Interviewee J said "...the system worked a little bit different in production, because of a new environment, and since we had simulated backend data we got [a few bugs] back..."
6 Analysis & Discussion

This chapter starts with an analysis of the pilot study, which is followed by two separate analyzes of the two case studies. At the end of the chapter, a cross case analysis is presented followed by a discussion. Parallels between the literature study and result in chapter 5 are made.

The empirical results, described in chapter 5, are analyzed and discussed in separate sections. Section 6.1 gives an analysis of the pilot study. This is because it was the base for the multiple case studies. In order to understand how the questions for the multiple case studies were prepared one needs to have access to the same information as the researcher had before conducting the multiple case studies. Section 6.2 gives an analysis of the empirical results Project 1. Section 6.3 gives an analyze of the empirical results from Project 2. In section 6.4 the results and analysis from both case studies are combined in order to be analyzed so that general conclusion can eventually be drawn.

6.1 Pilot Study

6.1.1 Project Management

From the interviews it is clear that they strive to follow DAD, but they do not manage. They are aware of the weaknesses of other agile methods and therefore try to use DAD. A big issue here is that very few actually know what DAD is and when comparing the roles in a Dewire’s project teams, see Appendix E, we see a difference from the traditional DAD roles. However, this might be due to the terminology which the customer is comfortable with. Another issue is that DAD best suits team with fewer than 15 team members who are close geographically, which does not fit for all projects at Dewire.

What distinguishes DAD from several other agile methods is the inception and transition phase. Since the observations are limited to iterations after the inception phase, but before the transition phase no clear conclusion in regards of which method the company actually uses can be made. Due to the answers from the interviewees and from the observation I would agree with interviewee 4 that they are using DAD, ad hoc style. Using an ad hoc approach in itself leads to uncertainties.

There exists a discourse between management and developers. The management team thinks that they have retrospectives after every iteration, but this was contradicted by the developers. However, these retrospective seems to be quite good and all interviews thought of them as beneficial so maybe Dewire should have them after every iteration. Only project managers mentioned that they had futurespectives, while none of the developers mentioned it. This might be due to the similarity between a retrospectives and futurespective making it hard for the team members to differentiate between them. However, this might be an issue since it might mean that the purpose of them is unclear.

6.1.2 Technological Issues

The technological issues were an uncertainty before, but have now over time become a risk, which they have worked on to handle. By having a station which they have setup on beforehand they reduce
the time needed to setup everything. From the pilot study no big issues in regard to technology have been identified in the current state of Dewire.

During the pilot study no effort was put into the existing technological infrastructure, architecture and APIs of the case company. However, these might be a source of uncertainty and therefore the conclusion that no technological uncertainties exist can not be made.

6.1.3 Documentation & Communication

From the interviews it seems like it is unclear what they document and how it should be documented. This might be because they lack a formal procedure for documentation. This results in knowledge and information becoming implicit rather than explicit. This results in teams having key members which becomes invaluable.

From the pilot study it was concluded that an “Us vs Them” feeling exists. However, this seems to have been a bit subsided to the fact that team members are able to meet, due to the geographical distance. In context the "Us vs Them" might also represent roles, for example, one project has all the developers in Sundsvall and the "Them" refers to developers as a role, rather than the individuals in Sundsvall.

From the interviews it seems like the level and means of communication varied based on an employee role and geographical location. The developer said that they mostly used face to face communication which might be because all developers on that project are in Sundsvall at the moment, hence coding decisions will be made there. At the same time a team member in Stockholm might be the only member in the remote team, thus making it impossible to have face to face communication.

Interviewee 6 explained how the team had handled requirement uncertainty. The requirement uncertainty surrounding the project was that the customer do not know what they wanted. In this case it resulted in the customer being displeased with the product. However a way to deal with this was also identified which was keeping all decisions documented. This shows that it is possible to work proactively against uncertainties. Still not all projects keep such a journal.

All interviewees agreed that there could and should be more documentation, hence they are aware of their faults and have ideas on how to fix them. At the same time no effort is made to implement the improvement ideas. This might be due to lack of responsibility and since nobody is in charge of documentation it might get lost in the daily work.
6.2 Project 1

Table 7 shows which types of uncertainties that have been identified and what area they belong to. A frequency count is also displayed as well as whether the mentioned uncertainties have been observed by the researcher. A big issue which uncertainties may stem from is the ad hoc agile practices which the team follows. It was a consensus between the team members that they had retrospectives after every sprint, previously. At the moment they have them roughly 3-4 times a year and they state that this is because the project is more stable now and they do not need to have retrospectives as often anymore. During the research it has also been observed that sometimes daily stand-ups are canceled due to unknown reasons, this can results in a decrease in communication and also efficiency for the team members in Stockholm, since they are not actively working while waiting for the stand up call. No market uncertainties were identified for this project, this might be due to the nature of consultancy projects.

<table>
<thead>
<tr>
<th>Area</th>
<th>Uncertainty</th>
<th>Freq.</th>
<th>Obs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environmental</td>
<td>Customers are not agile</td>
<td>7/7</td>
<td></td>
</tr>
<tr>
<td>Uncertainty</td>
<td>External system have bugs and not sufficient</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>documentation</td>
<td>4/7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Internal customer requirement ambiguity</td>
<td>4/7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Customer don’t have sufficient resources</td>
<td>3/7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GDPR</td>
<td>2/7</td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Ambiguous tasks</td>
<td>5/7</td>
<td>X</td>
</tr>
<tr>
<td>Uncertainty</td>
<td>Task dependencies</td>
<td>4/7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hard to estimate tasks</td>
<td>4/7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Testers communicate with developers if unclear story</td>
<td>3/7</td>
<td>X</td>
</tr>
<tr>
<td>Socio-Human</td>
<td>Not constant communication between the offices</td>
<td>7/7</td>
<td>X</td>
</tr>
<tr>
<td>Uncertainty</td>
<td>Slack messages are sometimes ignored</td>
<td>4/7</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Remote members lack information</td>
<td>4/7</td>
<td>X</td>
</tr>
<tr>
<td>Resource</td>
<td>&quot;Specialist&quot; members</td>
<td>6/7</td>
<td>X</td>
</tr>
<tr>
<td>Uncertainty</td>
<td>More resources than the customer wants</td>
<td>1/7</td>
<td></td>
</tr>
<tr>
<td>Requirement</td>
<td>Requirement quality varies</td>
<td>5/7</td>
<td></td>
</tr>
<tr>
<td>Uncertainty</td>
<td>Mid sprint requirements</td>
<td>4/7</td>
<td></td>
</tr>
<tr>
<td>Technological</td>
<td>Test and production environment are different</td>
<td>1/7</td>
<td></td>
</tr>
<tr>
<td>Uncertainty</td>
<td>New operating system needed a completely new app</td>
<td>1/7</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Overview of identified uncertainties for Project 1

6.2.1 Environmental Uncertainty

As we can see from table 7 project 1 was exposed to multiple environmental uncertainties. However, four out of five of the environmental uncertainties were based on the customer. The level of agility
for a project is limited by the agility of the customer (Ambler and Lines, 2012). Even though they have representatives from the customers who watch demos and communicate on a regular basis, they are not agile. The team is working with a component of an entire solution outsourced to several consultancy firms. Most parts of the solution do not affect Dewire’s component, thus reducing the number of uncertainties which might occur. However, Dewire uses one external system for their project which have sub-optimal documentation. The team members have commented on the difficulties in understanding the documentation when working showing that it can be improved. Once bugs had been identified it could take a while, over a month, before it was solved. This could be a result of the fact that the customer is not agile. Once the customer have been notified it might follow the traditional waterfall approach before the bug fix is released into production environment, thus the process can be seen as stiff and slow by an agile team.

As mentioned in section 5.3 the customer also have offices at multiple location and two of those being in Sundsvall and in Stockholm. They are in a similar situation as Dewire, with employees related to the project at both locations. This results in Dewire being affected to uncertainties stemming from the customers partially distributed teams. Just like Conchuir et al. (2005) mentions, there seem to be a lack of a shared vision and goals among the members of the customer team. This presents itself, by internal requirement ambiguity. For Dewire this means that requirements might be interpreted different in Sundsvall, which is where they are made, and in Stockholm, which is where they are tested.

As mentioned by interviewees D, E and G team’s testers are used by the customer for acceptance tests. This an opportunity which can be exploited by the team. If a task does not pass the acceptance test, then Dewire’s testers can contact the team immediately and notify of incoming bug fixes, rather than them taking the team by surprise.

GDPR was the only outcome of environmental uncertainty not related to the customer. For this project GDPR has been classified as a risk, which they have mitigated. For this specific project GDPR only affected the crashlytics software used by the apps according to interviewee E. Hence, GDPR was an uncertainty classified as threat with minor consequences for the project, with planning starting four months before coming to affect. The few number of interviewees who mentioned this indicates that the effects of GDPR have not been crucial for the project.

As mentioned, the source of most of the environmental uncertainties stems from the customer. Most of the uncertainties are threats. However, it is possible for the project to turn insufficient customer resources into opportunities. Based on good communication with the customer it is possible for the team to handle threats and exploit opportunities when they arise. This corresponds with the study of Waldman et al. (2001) who found how a socio-human factor could affect projects during high levels of environmental uncertainty. This shows that it might be possible for socio-human factors to affect the outcome of uncertainties.

### 6.2.2 Task Uncertainty

Task uncertainties had the second most number of uncertainties. 5/7 of all interviewees stated that there are a few unclear stories. Ambiguous and unclear tasks result in an unclear solution, thus it is important to have a shared goal for the project (Dönmez and Grote, 2018). Since partially distributed teams are vulnerable to a lack of shared vision and project (Conchuir et al., 2005). This
shows that the project is very vulnerable to task uncertainties at the moment, techniques such as daily stand-ups, sprint planning and retrospectives are used to reduce threats. Agile tools are therefore of great importance when dealing with task uncertainties, this is why it is problematic that the team do not have retrospectives and sometimes cancels the daily stand ups with no warning. At the moment the team members handle most ambiguous tasks by talking with the lead developers.

A result of the unclear tasks are that the testers do not understand the tasks and therefore do not know what to test. To solve this the testers contact the developer and asks what should be tested. This can result in the wrong thing being tested and then during acceptance testing the task will be sent back to the team. In order to avoid this, the team should not allow the testers to communicate with the developers regarding ambiguous tasks, this also forces the developer to document all changes and decisions in JIRA. The testers role is to verify that the code fulfills the tasks, not that it executes.

Another task uncertainty that was brought up was the problems with estimating tasks. In the past they had a tendency to underestimate the completion time, resulting in the team being stressed at the end of a sprint and hurrying with the tasks. While working with the project the estimations have become better, thus not posing a great threat at the moment. One of the sub teams had overestimated the completion time of the tasks. For the team this was an opportunity, as it allowed them to take tasks from the next sprint and complete them. The team leads and the architect begin with the sprint planning roughly two weeks before the actual sprint. This planning includes turning requirements into tasks, prioritize them and giving a rough estimate of the completion time. This allows the team to pick tasks from a future sprint when completion time of tasks have been overestimated.

Since the team have chosen to create two sub teams communication and planning between these sub teams are of great importance. The division has created the two sub teams front end and back end. The front end systems should be able to send data to the back end as well as retrieve it. This results in task dependencies where the back end and front end should be able to communicate. The task dependencies affect the front end team more since external programs can be used to mimic the behaviour of REST requests and can see how data sent is being handled. If the front end team are dependent on a smaller GET request it is possible for them to create test data which is retrieved, rather than the actual data. However, other requests such as PUT, DELETE and POST are harder for the front end team test. The task dependency between front end and back end tasks are viewed as a threat within the project and in order to reduce these threats they either plan so that the back end part of the task is higher prioritized or even done the previous sprint. One of the results of task dependency is that the team members need to switch tasks while waiting for a part to be finished, this results in switching costs which lowers the efficiency.

6.2.3 Socio-Human Uncertainty

All interviewees were aware that the communication between Sundsvall and Stockholm is not constant. Communication is used as a tool to increase awareness (Rodríguez et al., 2010). The lack of informal communication can result in a more negative relationship among the team members (Conchuir et al., 2005). The lack of communication is probably what results in the ”Us vs Them” feeling which exists at Dewire and it can be hard to trust the team members. By not having constant communication with all team members a shared vision might not be created, thus increasing the
negative affects of task uncertainties.

Another issue brought up by four interviewees is that Slack messages are sometimes ignored. The reason behind this is that they do not want to be disturbed. It is especially during meetings and when team members are in the middle of something that they do not notice and respond to the messages directly. The possibility to ignore direct messages can be seen both as an opportunity as well as a threat. The opportunity stems from the fact that team members who are working with something and need their full concentration, can work without being disturbed, thus increasing their efficiency. When everybody is located at the same office people will instead contact each other face to face, forcing each other to leave tasks in order to communicate. The threat stems from the fact that the remote team members might feel that they are being ignored. When messages are being ignored it is important that the team has a lot of trust for each other. However, it is common for the remote members to have trust issues as they might not have the same sense of belonging as the co-located members do (Webster and Wong 2008).

A majority of the interviewees also believed that the remote members lack information. This is a threat which stem from lack of constant communication and probably not from daily stand ups all the time. All interviewees agreed that informal communication about the project takes place, thus the members who miss out on this communication also misses information. The team is supposed to document all decisions in JIRA, but since this is not the case, the information might never reach the remote members. The lack of information results in tasks being hard to understand and resulting in the testers, the remote members of the back end team, to ask the developers for instructions. However, it is possible to turn this uncertainty into an opportunity. This can work as verification for documentation, if the tester can not understand what should be tested from the story on JIRA, then the developers need to add comments and documentation to the story. This will allow more tasks being accepted by the customer since there is a decrease of ambiguous tasks.

The time spent at the Dewire office have showed that face to face meetings are highly valued for this project. Every week the remote members visits Sundsvall and get to spend time with the co-located team. By spending more time with the team natural communication can take place, thus leading to an increase in trust (Conchúir et al. 2005). The trips to Sundsvall and back have negative side effects. A 3,5 hour trip are 3,5 working hours where the remote members are not working efficiently. It is important to weigh the downsides of the trips to Sundsvall with the benefits.

The socio-human uncertainties listed in table 7 stem from the partially distributed structure of the team. The consequences of lack of communication are strengthened by not having daily stand-ups and retrospectives after every sprint. The lack of communication can also be used as verification tool to ensure that the documentation is sufficient enough for the remote members to understand, meaning that external stakeholders will understand it.

6.2.4 Resource Uncertainty

For this specific project, the team members only identified threats. The reason behind this is unclear, but corresponds with the findings of Dönmez and Grote (2018), who found that team members more easily spot threats than opportunities.

The members of the teams can be seen as specialist. The testers are testers, the front end developers
specializes in either web, android or iOS. This can be problematic since the team only have one web, one android and one ios developer. If one of these developers was sick or had an unplanned leave this would cause huge consequences for the project. If the all front end members were to share knowledge if would be possible to cover for a loss of human resource, rather than one of the applications not progressing at all during the sprint. This also goes along with the DAD methodology which advocates that a team should consist of generalists with a breadth of knowledge \cite{Ambler2012}.

The project manager also stated that the project uses more resources than the customer intends. This can be viewed as a threat due to funding cutting of the possibility of additional resources. However, this threat could be mitigating by communicating the needed resources with the customer and making them aware that the project needs more resources.

6.2.5 Requirement Uncertainty

The requirement uncertainties are problematic for the project as they result in other uncertainties and can pose a huge threat for the team. The quality of the requirements have varied. Poor requirement quality can lead to task uncertainties. What the team wants from the customer are use cases which explains how product should work, not ambiguous one-liners. Interviewee A even stated that the demands are created iterative with the customer in some cases. For this specific project the architect has taken on the role to discuss requirements with the customer in order to get a clearer understanding of the ambiguous requirements, as recommended by Ambler and Lines \cite{Ambler2012}.

When working agile new requirements are always welcomed and expected from the customer. Usually these requirements are handled in sprint planning and then implemented during the actual sprint. However, sometimes requirements made in the middle of a sprint might affect the tasks the team are currently working on. This in itself is an uncertainty which the project has no standard procedure to handle. All interviewees agreed that smaller changes can be added to the current sprint without any trouble, since one of the project goals is to please the customer. However, it was hard for the interviewees to describe what is classified as a small change. In regards to bigger changes no consensus existed among the interviewees. A majority stated that for bigger changes, the new requirement will have to be added for the next sprint. Interviewee A on the other hand stated that if a new requirement affects the current sprint you break the sprint and begin to implement the new requirement. As mentioned earlier mid sprint requirements are to be expected, but since no clear way to handling them exists they have become a threat for the project.

6.2.6 Technological Uncertainty

Fewer technological uncertainties were identified. The technologies used for the project are described as "non exiting" and "standard" by interviewee F. If comparing with the scale by Dvir and Shenhar \cite{Dvir2007} the project can be considered as low tech since, the level of technological uncertainty is low. Interview G stated that the testing environment and production environment differ. This in itself is a risk which the team are aware of, but the consequences of this risk is hard to measure. A known consequence is bugs occurring when the solution is released, causing the team to take on maintenance responsibility. Another uncertainty which was brought up by interviewee F is that they had to redo the entire watchOS app, due to an upgrade of the watchOS. The app was first created for watchOS 1 and when upgrading to watchOS 2, it had to be redone.
6.3 Project 2

Table 8 shows which types of uncertainties that have been identified and what area they belong to. A frequency count is also displayed next to the uncertainty. No market uncertainties were identified for this project, this might be due to the nature of consultancy projects.

<table>
<thead>
<tr>
<th>Area</th>
<th>Uncertainty</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socio-Human</td>
<td>Ignored direct messages</td>
<td>2/6</td>
</tr>
<tr>
<td>Uncertainty</td>
<td>Poor communication within team</td>
<td>2/6</td>
</tr>
<tr>
<td></td>
<td>Stockholm members missed a few stand ups</td>
<td>2/6</td>
</tr>
<tr>
<td></td>
<td>Stockholm office continued working after project was done</td>
<td>1/6</td>
</tr>
<tr>
<td>Resource</td>
<td>Key team members</td>
<td>5/6</td>
</tr>
<tr>
<td>Uncertainty</td>
<td>Hard to find documentation</td>
<td>3/6</td>
</tr>
<tr>
<td></td>
<td>Project lost resources for two weeks</td>
<td>2/6</td>
</tr>
<tr>
<td></td>
<td>Two project managers</td>
<td>1/6</td>
</tr>
<tr>
<td>Environmental</td>
<td>Customer involvement</td>
<td>5/6</td>
</tr>
<tr>
<td>Uncertainty</td>
<td>Faulty guidelines for external system</td>
<td>2/6</td>
</tr>
<tr>
<td></td>
<td>Project suddenly stopped</td>
<td>2/6</td>
</tr>
<tr>
<td>Task</td>
<td>Task modularisation</td>
<td>5/6</td>
</tr>
<tr>
<td>Uncertainty</td>
<td>Ambiguous tasks</td>
<td>3/6</td>
</tr>
<tr>
<td>Requirement</td>
<td>Unrealistic requirements</td>
<td>2/6</td>
</tr>
<tr>
<td>Uncertainty</td>
<td>Last minute requirements</td>
<td>1/6</td>
</tr>
<tr>
<td>Technological</td>
<td>Test and production environment differently</td>
<td>1/6</td>
</tr>
</tbody>
</table>

Table 8: Overview of identified uncertainties for Project 2

6.3.1 Socio-Human Uncertainty

Two team members stated that the communication was poor within the team, it was also these two members who said that they believed that direct messages were ignored. The two team members mentioning this were located in Sundsvall and therefore not as used to working in partially distributed teams. The Stockholm members are more used to this since they have fewer people in the office and do not have the necessary resources in the office. Since Dewire have no formal training for the partially distributed teams it can be hard for a member with little or no experience to join. Hiltz et al. (2016) recommends training being a part of partially distributed teams since it gives a common ground for all members.

Another issue for the project was that the Stockholm office missed a few daily standups. The reason behind this is unclear, it might have been a mix of technological issues as well as a miss in communication. Missing daily standups can be very problematic for partially distributed teams as it is the main way to get an overview of the project and communicate regarding the different tasks. Daily standups are supposed to give the team the possibility to discuss current tasks and update completion times of tasks, therefore it is important for all team members to participate.
One big discourse existed between the offices in Stockholm and Sundsvall. The project was supposed to stop implementing new features and the solution had been given to the customer. However, the Stockholm office continued working with new requirements from the customer. From the interviews it has been unclear what actually happened, but this still shows that the communication between the two offices is not clear enough. This miscommunication could be the result of one of two possibilities: either the Stockholm office did not receive information about the completion of the project, or Sundsvall did not receive information regarding an extension of the project.

The socio-human uncertainties listed in Table 8 stem from the partially distributed structure of the team. The consequences of lack of communication are strengthened by not having daily stand-ups. The lack of communication can also be used as a verification tool to ensure that the documentation is sufficient enough for the remote members to understand, meaning that external stakeholders will understand it.

6.3.2 Resource Uncertainty

Having key team members, or members with a lot of knowledge and experience of the project can be viewed as both a threat and an opportunity. When losing these members the team can reach a standstill and might find it hard to make progress. But by having these members it might be easier to interpret requirements from the customer. In this case the key members had been part of an earlier generation of the project, thus they had a background knowledge of the project and had a clear view of the purpose of it.

The two most mentioned resource uncertainties are related. By having key team members with lots of knowledge it will be easier for a team member to ask for help rather than find it, through documentation. The team members were aware that documentation existed, but it was hard to keep track on how and where to find it. By having key team members the team makes it possible for the documentation management to be careless and sub-par.

The project was not top priority at Dewire. This resulted in the team losing its members and no progress could be made. There is no way to avoid this situation, hence not much thought was given to this problem by Dewire. But this made the schedule pretty hectic and may have been the reason why interviewee I felt like they had to work overtime to complete all tasks. However, even if resources are pulled mid iteration, the team should strive for it not affecting the progress (Conboy et al., 2012).

Interviewee H was one of the project managers of the project; the other project manager was located in Stockholm and focused more on the customer. The reason for having two project managers was that this was interviewee H’s first project at Dewire. However, this also caused an issue regarding the work division. The project managers were not aware who communicated with the customer and both probably contacted the customer. This shows that the project used more resource needed than necessary. When having too many resources for a project it might turn to be more ineffective rather than the opposite.
6.3.3 Environmental Uncertainty

The biggest issue was the lack of communication with the customer. According to interviewee H this was an issue brought up by all members of the team during retrospectives. The customers were not invested in the project and they were absent for most of it. This resulted in requirement uncertainty being hard to handle since no clarification of unclear requirements could be established. The low involvement of the customer also resulted in resource being pulled from the project. However, in this case the consequences affected the customer more than the team.

Two team members mentioned how the project came to an abrupt halt. The customer decided to cancel any further development of the software and only have maintenance of it. This became known to the team mid sprint and thus Dewire have half finished software not being used. This shows that it is important to predict environmental factors, as stated by Hoque (2004). The communication between Dewire and the customer should have been clearer and Dewire should have had more demands on the customer.

For this project the greatest environmental uncertainties stemmed from the customer and communication with the customer. The other environmental uncertainty were the faulty guidelines for the external system being used. However this would not have been a huge problem if the communication with the customer was better. This corresponds with Waldman et al. (2001) who found that socio-human factors can be used to handle environmental uncertainty.

6.3.4 Task Uncertainty

Most team members stated that the geographical distance did not create a huge problem, since the work division was clear. In other words, a clear modularisation of tasks existed. The modularisation allows the team to work in parallel, but it also prevents knowledge sharing (Ambler and Lines, 2012). The modularisation of tasks also makes it possible to reuse fragmented code later on in the project as well as for other projects.

The team also mentioned that ambiguous tasks were an issue. As they explained it, these ambiguity stemmed from ambiguous requirements. The way the team handled these ambiguities was to talk with more senior members, the key team members, of the project who had been a part of it for a longer time. They strove to ask the project manager for a clarification, but due to the poor customer relation it took a while to get a clarification.

6.3.5 Requirement Uncertainty

The requirements from the customer were not thought through and it was clear to the team members that the requirements came from a person with little technical knowledge. This unrealistic requirements resulted in ambiguous requirements, which in turn resulted in ambiguous tasks. The DAD method emphasizes the importance of customer, or stakeholder, relation and are aware that a lack of communication between the team and the stakeholders result in requirement uncertainties which are hard to handle (Ambler and Lines, 2012).

At the end of the project the project manager became aware of the documentation requirements.
These were not stated explicitly at the beginning, thus creating a hurdle for the team. By having documented during the process this did not affect the team, according to interviewee H. However, interviewee I claimed to have worked overtime, which contradicts the statement of interviewee H. As stated by half of the team members, the documentation already existed, but it was not clear where it was stored.

6.3.6 Technological Uncertainty

The project could be considered as low-tech since the technologies being used were well established at the time (Dvir and Shenhar, 2007). The one technological uncertainty mentioned was that the test and production environment works differently. This not an uncertainty specific for either agile or partially distributed team, but rather for software development projects. Creating test environments are costly and even if a well established test environment lack real users.

6.4 Cross-Case Comparison

Even though the two teams mentioned a few opportunities, a majority mentioned were threats. This corresponds with what Dönmez and Grote (2018) found during their research on uncertainties. The concept of threats, as well as risks tend to be more well known within a project and therefore the focus lies mitigating them. Hence, the effects of uncertainties are mostly viewed as something negative which may delay the project or not completing all tasks during the sprints.

![Figure 4: Frequency of threats and opportunities mentioned](image)
As we can see from figure 4 extremely little focus were given to opportunities. Those uncertainties classified as "Both" are those mentioned in a negative manner by the interviewees, but which can be used as an opportunity if clear guidelines exists. The differences between the frequency of threats and opportunities shows that Dewire need to learn how to identify opportunities.

6.4.1 Uncertainties

![Figure 5: Number of uncertainties identified](image)

The different types of uncertainties identified differ among the two projects. As we can see from figure 5 only five uncertainties were shared between the two projects. One of the bigger issues with the uncertainties was that no clear method to handle them existed which results in possible opportunities turning into threats. The idea of agile methods are to allow and manage uncertainties in order to mitigate threats as well as exploit opportunities.

The most common type of uncertainty between the two projects were environmental uncertainty. However, these uncertainties are not specific for partially distributed teams, but could exist for any agile software development team. No market uncertainties were identified by any interviewee. This probably stems from the fact that Dewire is an IT consultancy firm and neither of the two projects were B2C, rather B2B. The market uncertainties affecting the project might have been identified by the customer who in turn came with new requirements, thus the market uncertainties could be viewed as requirement uncertainties for these projects.
The socio-human uncertainties for both projects stemmed from that the teams were partially distributed. When having partially distributed teams explicit information and knowledge and transparency within the team becomes even more important. It becomes clear how informal communication affect the project and if the decision made during the informal communication, the remote members have no way of getting the information.

The differentiation between task and requirement were hard for both of the projects. Team members had a tendency to confuse tasks with requirements and vice versa. It seemed like requirement uncertainties resulted in task uncertainties, especially those related to ambiguity. The requirement uncertainties were a direct result from customer relation, but the customers action resulted in environmental uncertainty.

When working with projects resource uncertainties may affect the progress greatly. Every week the project managers needed to show why they needed certain people for their project. Project 2 can not afford to lose its key team members, thus the consequences of them leaving for another project increased. Project 1 did not have these ”key team members” but most members can be viewed as specialists rather than generalists. If we are to look at the front-end team each developer were in assigned to one platform. If one of these members would have been sick, than their platform would not have made any progress.

6.4.2 Managing Uncertainties

During the research no method for handling uncertainties have been identified at Dewire. Dewire try to use DAD as way to increase their flexibility, but not all project members are comfortable with the method and some do not even know how DAD differs from other agile methods. A great way to increase the teams flexibility would be to teach all members how to both identify and handle uncertainties. At the moment it is up for the individual team members to take the responsibility of managing uncertainty. This results in uncertainty being managed differently for projects and the effect of uncertainties varies greatly.

Partially distributed teams have it hard to work according to standard practices since the different location might have different practices. The Stockholm members believed that they followed practices more suitable for partially distributed teams and that they could handle the lack of face to face meetings and informal communication better than the Sundsvall office.

As we can see in table 9 methods to handle uncertainty exists. Many of the tools in the DAD method can be used to mitigate threats, exploit opportunities and identify threats. The concept of opportunity might be harder to grasp due to the importance of mitigating threats. During the interviews the team members seemed to put greater value into mitigating threats than exploiting opportunities.
<table>
<thead>
<tr>
<th>Action</th>
<th>Source</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face to face visits</td>
<td>Project 1 &amp; Conchuir et al. (2005)</td>
<td>By having regular visits, the remote members get an opportunity to take part in the informal communication and the team get to know one another better, thus increasing the trust.</td>
</tr>
<tr>
<td>Planning for uncertainty</td>
<td>Project 1</td>
<td>In order to manage uncertainties project 1 have decided to not allow implementation of new features the last week during the sprint. This is mainly to allow bug fixes, but it also gives the team time to identify and handle uncertainty.</td>
</tr>
<tr>
<td>Not constant communication between the offices</td>
<td>Project 1 &amp; 2</td>
<td>The result of this is that implicit information does not reach all members. This forces information and knowledge to be explicit instead. An opportunity which project 1 could could exploit is that the testers should be able to test the features based on documentation.</td>
</tr>
<tr>
<td>Sharing responsibility</td>
<td>Project 2 &amp; Ambler and Lines (2012)</td>
<td>What is meant by &quot;sharing responsibility&quot; is to allow more than one person being able to program on multiple platforms. For project 2 the developers shared responsibility of the Android and iOS app, thus reducing the effect of threat related to resource uncertainty.</td>
</tr>
<tr>
<td>Not allowing mid-sprint requirements</td>
<td>Project 1</td>
<td>The purpose of the sprints are to complete a set of tasks. If new requirements affecting the tasks are added during the sprint it may cause a project to not complete the sprint within deadline. This results in a stressed team and it makes it harder to prioritize the different tasks.</td>
</tr>
<tr>
<td>Task modularisation</td>
<td>Project 2 &amp; Dönmez and Grote (2013)</td>
<td>Task modularisation allows the team to make better estimations and it becomes easier to to distribute the tasks evenly. Creating small tasks makes it possible to create small modules which can be reused for later project. However, smaller tasks increases the importance to sort out task dependencies.</td>
</tr>
<tr>
<td>Daily standups</td>
<td>Project 1, 2 &amp; Ambler and Lines (2012)</td>
<td>Daily stand ups gives a great tool for identifying uncertainties. They can allow all members to become aware of the current situation, thus the responsibility of identifying uncertainties are moved from the individual to the team. Since the teams usually have a mix of newcomers and experienced members retrospectives are a great tool for the team for knowledge sharing. All team members saw the value of the retrospectives, but believed that they could manage without them. Retrospectives gives a platform to discuss the identification of opportunities in the previous sprint and how they were handled.</td>
</tr>
<tr>
<td>Retrospectives</td>
<td>Project 1, 2 &amp; Ambler and Lines (2012)</td>
<td>Daily stand ups gives a great tool for identifying uncertainties. They can allow all members to become aware of the current situation, thus the responsibility of identifying uncertainties are moved from the individual to the team. Since the teams usually have a mix of newcomers and experienced members retrospectives are a great tool for the team for knowledge sharing. All team members saw the value of the retrospectives, but believed that they could manage without them. Retrospectives gives a platform to discuss the identification of opportunities in the previous sprint and how they were handled.</td>
</tr>
<tr>
<td>Ignoring slack messages</td>
<td>Project 1 &amp; 2</td>
<td>At a more general level ignoring slack messages are not a good idea and is probably a threat. But if clear guidelines exist it gives an opportunity for team members to get into a flow. By ignoring slack messages and requests you reduce the amount of task switching which is needed, thus decreasing efficiency. The low level design meetings are held by the architect, team leads and customer. They allow the team to create a backlog of prioritized tasks for the upcoming sprint. This allows the team to take tasks for future sprints if they were completed faster than expected.</td>
</tr>
<tr>
<td>Action</td>
<td>Source</td>
<td>Effect</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Documentation guidelines</td>
<td>Project 1, 2 &amp; Ambler and Lines (2012)</td>
<td>From both projects it is clear that documentation guidelines need to exist. There needs to be a consensus of what is added to JIRA, and if it the information can not be found on JIRA where can it be found. Documentation can be used as a tool to facilitate knowledge sharing and if information is not easy to find knowledge which is supposed to be explicit becomes implicit instead.</td>
</tr>
</tbody>
</table>

Table 9: Overview of how uncertainties are managed

6.4.3 Sustainability

In Stockholm over 50% of all domestic flights are business trips and a total of 453233 tons of CO₂ is released by all business trips originating from Stockholm (Porssö, 2017). Well managed partially distributed teams can use visualization tools and communication tools instead of face to face meetings and communication. When deciding to have partially distributed teams or distributed teams the company needs to realize that making frequent trips to facilitate meetings might not be environmentally viable.

Social sustainability focuses on how business can impact the life of people (United Nations Global Impact). The rights of an individual should be respected and their needs should be fulfilled. Constant travelling can reduce the life quality of the team members since their exists value in having a work and personal life balance. The "Us vs Them" feeling which is the result of some of the socio-human uncertainties needs to be reduced as it lowers the well being of the team members. When the teams did not know how to handle uncertainties, some members felt that it caused stressful situations. By implementing uncertainty management the stress can be reduced.

Using partially distributed teams allow for a more evenly resource distribution within the organization, thus creating long term economic growth. If uncertainties are not managed or handled ad hoc it leads to a greater risk of project failure. The risk of project failure increases due to not mitigating threats and not being able to exploit opportunities. Thus, the flexibility which stems from agile practices results a higher economic sustainability.

6.5 Discussion of Research Methods Used

The purpose of this study is complex and many underlying factors and variables exists. For this reason the use of multiple case studies were used, more about the methodology can be found in Chapter 4. However the chosen methods results in limitations to the study and the accuracy of the results found. When choosing a qualitative method rather than a quantitative a loss of generalizability is followed. By conducting two case studies the result can be hard to generalize, especially when both cases are from the same company. The context of the two cases affects the generalizability of the study. The context of the two cases limits the research to only be applicable for partially distributed teams without any cultural distance or in different time zones. The geographical distance between the two offices is also limited, thus regular visits are possible which isn’t the situation for all
Since no comparison has been made with either co-located teams or distributed teams it can be determined which uncertainties stem from the structure of partially distributed team. However, during the analysis phase the source of each uncertainty were evaluated trying to understand whether they stemmed from the geographical distance between team members.
7 Conclusion

This chapter concludes the analysis with summarizing my findings and how they answer my research question, as well as the sub research questions. This is followed by the implications of this study and ends with ideas for future research.

7.1 Conclusion

The main purpose of the study was to identify how uncertainties affect a partially distributed team. To fulfill the purpose the main research question was formulated as:

*How are partially distributed agile software development teams affected by uncertainty?*

The main research question was divided into two sub-questions in order to find which types of uncertainties existed and how they were handled. The effects of uncertainties depended on how they were viewed by the team and how they then decided to handle them. This study showed that uncertainties are mostly correlated to threats and therefore focus lies on mitigating them. The effects of the uncertainties, tends to be viewed as consequences resulting in not finishing all tasks during the current sprint. The answer to this questions was severely limited by only studying two different cases and no generalization could be made for this questions.

7.1.1 Research Question 1

The first research question was formulated:

*What areas of uncertainty are partially distributed agile software development teams typically subject to?*

In order to answer [RQ1] general areas of uncertainty needed to be identified and defined. These were:

- Market Uncertainty which relates to how the market perceives the product.
- Requirement uncertainty which stems from unclear or changing demands from relevant stakeholders.
- Socio-human uncertainty which relates to knowledge sharing and the relationship between team members.
- Resource uncertainty which stem from incomplete information regarding availability of all resources needed.
- Environmental uncertainty which stems from action made by external players.
- Task uncertainty which relates to ambiguous tasks or outcomes.
- Technological uncertainty which is determined by the maturity of the technology being used.

After identifying these uncertainties and comparing them to those found during the case study we could see that agile partially distributed teams are subject to are: technological, requirement, resource, task, socio-human and environmental. Environmental uncertainties had the highest number
of identified uncertainties, but the socio-human uncertainties stemmed from that the teams were partially distributed. Hence, this study showed that partially distributed teams lead to socio-human uncertainty.

7.1.2 Research Question 2

The second research question was formulated: 
*How do partially distributed agile software development teams handle uncertainties?*

The research question was answered using the result from the two case studies and the literature review. How the two projects chose to handle uncertainties can be found in table 9. The research found that how partially distributed teams handle uncertainties are based on the level of experience of the team members. Many of the identified actions to handle uncertainties are based on agile practises showing that agile methods should be implemented fully in order to handle uncertainties.

7.2 Implications

The implications for software development teams refer to the practical use of the findings in this study, whereas the academic implications refer to the theoretical use. I want to point out that the implications should be considered lightly due the limitations of the study.

7.2.1 Software Development Teams

In the practice this research shows the value of agile principles. It shows that software development teams needs an established way to deal with uncertainties otherwise they will turn into threats which can result in project failure. Even though agile methods focus on share responsibility the team should not forget the aspect of knowledge sharing. If knowledge is not shared between the team members it makes it hard for new team members to learn how to identify and handle uncertainties. A clear standard on how to handle uncertainties needs to be established, this is even more important for partially distributed teams since they have a tendency to lack common practices. When teams lack common practices and do not follow the agile method it easily results in an ad hoc approach which have the highest failure rate of all management methods for IT projects (Ambysoft, 2013).

When choosing to have a partially distributed team Dewire needs to weigh the negative aspect with the positive one. Such teams creates an “Us vs Them” feeling and the lack of informal communication affects the project and team negatively. When creating partially distributed teams it is important to have members experienced working in such teams. All team members also needs to realize that documentation becomes of great importance when not all members are co-located in order to facilitate knowledge sharing.
7.2.2 Academia

There is a lack of consensus regarding uncertainty. This research could help gather the different definitions and combine them into one. If this is not the case, it still shows that a consensus needs to be established in order to research uncertainty more deeply. If the term uncertainty does not have a clear definition it will be hard for researchers to continue on others work. This research also shows that the good outcomes of uncertainties needs to be valued. One should not focus on mitigating uncertainty. This research also showed that the geographical distance is a hinder for a team, which corresponds with literature described in section 3.5.

7.3 Future Research

During my research on the topics of uncertainties and partially distributed teams I have identified a few possible ideas for future research.

- Similar cases: The two case study are not enough to generalize the findings, especially since the two cases were from the same company. In order to verify my findings a study on a different company, with a similar context needs to be conducted. Another method would be to use a quantitative study over multiple projects, thus being able to get a statistical generalizability. The researches by Dönmec and Grote (2018) and Marinho et al. (2017) were focused on the different types of uncertainties and they were conducted using multiple cases.

- Comparing with co-located teams and distributed teams: To find out which uncertainties are directly related to the partially distributed structure a comparison study could be made. By doing this it will be possible for a project manager to identify possible strengths of the different team structure. Research comparing co-located teams, distributed teams and partially distributed teams exits (Hiltz et al., 2016; Webster and Wong, 2008; Conchur et al., 2005), but they do not focus on uncertainties.

- Creating a framework based on multiple cases: Once more cases have been studied it may be possible to create a framework on how to handle uncertainties. Such a framework should include how to identify uncertainties and determine whether they are opportunities or threats. The framework should then be evaluated in order to determine whether or not it is applicable for partially distributed teams.

- What is agile: During my research I have noted that the teams tried to work after the agile method DAD, but do not succeed. I believe that it would be interesting to research why teams fail to follow agile methods. The idea behind agile methods are to handle uncertainty, so when the methods are not followed the flexability may decrease.
References


57


leadership attributes and profitability under conditions of perceived environmental uncertainty.

of Planning Education and Research.

Watson, R. T. and Webster, J. (2002). Analyzing the past to prepare for the future: Writing a
literature review. MIS quarterly, pages xiii–xxiii.

munication and trust in naturally occurring project teams. The International Journal of Human

Fifth Biennial Conference of the International Research Network for Organizing by Projects. Held


Appendices

Appendix A

Interview Protocol for introductory interview

- Hur fungerar er konsultverksamhet (för det mesta), får ett team hos ert ett projekt eller skickas konsulter ut till kund?
- Är det samma team man arbetar i eller är det stor variation (skillnad mellan stockholm och sundsvall)?
- Hur många medlemmar brukar ni vara i ett team?
- Samma ”scrum” master eller roterar det?
- Hur långa är de flesta projekten?
- Har ni sprintar?
- Hur långa brukar era sprintar vara?
- Hur nära arbetar ni med kund?
- Har ni någon form av riskhantering?
- Hur mycket tid läggs på dokumentation och vad tycker du/ni är nödvändigt att dokumentera?
- När ni blir satta på ett nytt projekt hur mycket tid läggs på planering?
- Hur ofta sker oväntade saker under en sprint? (ge något konkret exempel)
Appendix B

Pilot Study Interview Questions

Namn: 
Datum: 
Plats: 
Ålder

- Hur länge har du jobbat på Dewire?
- Vad gör du nu?
- Har det projektet något namn?
- Hur länge har du varit en del av projektet och hur länge har det hållt
  - Hur gick uppstartsfasen till (om delaktig)?
  - Hur var det att sätta sig in i projektet?
- Har du haft olika roller? Varför?
- Vilken arbetsmetod använder ni er av?
- Har ni dagliga standups?
  - Hur långa är de?
  - brukar de starta i tid?
- Har ni retrospectives efter varje sprint?
  - Är de givande enligt dig?
- Hur ofta blir det tekniska problem?
- Vad är teamets huvudsakliga kommunikationsmedel?
- Vilka roller finns i ditt team?
Appendix C

Case Study Protocol

A. Introduction to the Case Study and Purpose of Protocol

Chapter 1 states the scientific questions and the purpose of this study. In chapter 3 theoretical frameworks are described. The purpose of this protocol is to allow repetition of the study.

B. Data Collection and Procedures

See chapter 4.

At the Stockholm office my contact person was Petter Andersson who had the role as my supervisor during this research.

During my first visit to Sundsvall (13/2) Petter Andersson was my contact person at the location.

During my second visit to Sundsvall (27/3-28/3) Kristin Raukola was my contact person at the location.

C. Outline of Case Study Report

See sec 1.8.

D. Case Study Questions

See sec 1.4.
Appendix D

Color codes

- **Red**: Technology
- **Purple**: Resources
- **Green**: Requirements
- **Yellow**: Tasks
- **Blue**: Socio-human
- **Pink**: Environment
- **Brown**: Market
## Appendix E

### Agile Roles

<table>
<thead>
<tr>
<th>Project 1</th>
<th>Interviewee 1</th>
<th>Interviewee 2</th>
<th>Interviewee 4</th>
<th>Interviewee 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project Manager</td>
<td>Project Manger</td>
<td>Project Manager</td>
<td>Project Manager</td>
</tr>
<tr>
<td></td>
<td>Team Leads</td>
<td>Team Leads</td>
<td>Team Leads</td>
<td>Team Leads</td>
</tr>
<tr>
<td></td>
<td>Developers</td>
<td>Developers</td>
<td>Developers</td>
<td>Developers</td>
</tr>
<tr>
<td></td>
<td>Architect</td>
<td>Architect</td>
<td>Architect</td>
<td>Solution Architect</td>
</tr>
<tr>
<td></td>
<td>Test lead</td>
<td>Test lead</td>
<td>Scrum Master</td>
<td>Test Lead</td>
</tr>
<tr>
<td></td>
<td>Testers</td>
<td>Testers</td>
<td>Testers</td>
<td>Testers</td>
</tr>
<tr>
<td></td>
<td>Scrum Master</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project 2</th>
<th>Interviewee 4</th>
<th>Interviewee 6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project Manager</td>
<td>Project Manger</td>
</tr>
<tr>
<td></td>
<td>Testers</td>
<td>Testers</td>
</tr>
<tr>
<td></td>
<td>Developers</td>
<td>Developers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project 3</th>
<th>Interviewee 6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project Manager</td>
</tr>
<tr>
<td></td>
<td>Testers</td>
</tr>
<tr>
<td></td>
<td>Developers</td>
</tr>
<tr>
<td></td>
<td>Client Contact</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General Dewire</th>
<th>Interviewee 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project Manager</td>
</tr>
<tr>
<td></td>
<td>Solution Architect</td>
</tr>
<tr>
<td></td>
<td>Scrum Master</td>
</tr>
<tr>
<td></td>
<td>Product Owner</td>
</tr>
<tr>
<td></td>
<td>Developer</td>
</tr>
<tr>
<td></td>
<td>Test Lead</td>
</tr>
<tr>
<td></td>
<td>Testers</td>
</tr>
</tbody>
</table>
Appendix F

Interview protocol

The topics to be brought up during these interviews were:

**Project specific information**
**Role in project**
**Communication**
**Uncertainties**

In order to gather information regarding said topics these questions were used to start a conversational interview if more information regarding a topic was needed.

Describe the project.

What is your role in the project?

How does a typical sprint look like?

If the customer wants a highly prioritized change how long does it take until it’s ready for AT test?

If the customer changes their requirement mid sprint, how does it affect your work?

Are the tasks and stories clear?

What do you do if a task/story doesn’t make sense to you?

How much do you communicate with a) the tester, b) the developers, c) the members in Sundsvall, d) the members in Stockholm, e) the client, f) external people/organisations?

During your time on the project has anything unplanned happened?