



NTNU – Trondheim
Norwegian University of
Science and Technology

Developer and Customer Collaboration in Agile Software Development

Aleksander Torstensen

Master of Science in Computer Science

Submission date: June 2015

Supervisor: Maria Letizia Jaccheri, IDI

Co-supervisor: Konstantinos Chorianopoulos, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Preface

This thesis was written during a hectic period in my life. Including the obvious obligation of finishing my master degree, I am also the CEO of an award-winning startup company called Aalberg Audio.

My time as an entrepreneur has set me on a quest to seek out more customer and market focused ways of developing products. The trending business book Lean Startup by the Agile practitioner Eric Ries (2011) was my inspiration to look within the world of Agile software development.

This thesis is targeted to other entrepreneurs, software developers and project managers who are interested in finding best practices for delivering high-value products that actually fulfill the customer's need.

A special thanks to my Greek friend and supervisor Konstantinos Chorianopoulos for the interesting conversations and the guidance throughout the writing process. Also, thanks to the Department of Computer and Information Science at NTNU and its department head Maria Letizia Jaccheri, for the support.

Last, but not least, thanks to family, friends and loved ones for making me who I am today.

“Everything around you that you call life was made up by people that were no smarter than you. You can change it, you can influence it, you can build your own things that other people can use.” – Steve Jobs

A handwritten signature in black ink that reads "Aleksander Torstensen". The signature is written in a cursive style with a horizontal line above the last name.

Aleksander Torstensen

Trondheim, June 2015

Abstract

The contribution of this thesis is an in-field verification that Agile software development projects are prone to developer and customer collaboration problems. The focus of collaboration in Agile projects is a trending topic in literature, but there are still limited studies regarding stakeholder collaboration problems. This thesis is a research on the causes and consequences of collaboration problems between developers and customers in Agile software development projects. In addition, the thesis studies the contingency strategies that developers use to avoid collaboration problems with the customer. Empirical data was collected through interviews with the Norwegian software consultant company Bouvet, regarding their project with a governmental customer. Firstly, findings suggest that *lack of time commitment* and *governmental issues* was the core cause of collaboration problems. Secondly, the consequences of these collaboration problems were shown to be especially *low-value deliveries* from the developers and *problems in securing feedback* from the customer. Thirdly, it was found that developers used numerous strategies, including *e-collaboration*, to deal with collaboration problems with the customer. Overall, the customer had few signs of scepticism towards Agile, which has been indicated as a problem in previous literature. Even though Bouvet and their governmental customer show signs of close and trustworthy collaboration, the development team shows signs of deviations from key Agile principles. More studies on Agile projects, including different types of case projects and case companies, are recommended to enable more depth.

Table of Content

Preface	1
Abstract	2
Table of Content	3
1 Introduction	5
1.1 Research Questions	6
1.2 Context	8
1.3 Definitions	9
1.4 Method	11
1.5 Outline	11
2 Theoretical Framework	13
2.1 Agile Software Development	13
2.2 Types of Agile Methods	17
2.3 Collaboration	19
2.4 Collaboration in Agile Methods	21
2.4.1 Cause	25
2.4.2 Consequence	26
2.4.3 Contingency	27
3 Method	30
3.1 Participants	31
3.2 Interviews	33
3.3 Analytical Steps	34
3.4 Limitations	37
4 Results	39
4.1 Cause	39
4.2 Consequence	43
4.3 Contingency	48
5 Discussion	54
5.1 Cause	54
5.2 Consequence	57
5.3 Contingency	61

5.4 Summary.....	64
6 Implications and Further Research	66
6.1 Implications	66
6.1.1 Theoretical	66
6.1.2 Practical	67
6.1.3 Political	68
6.2 Further Research	69
6.3 Summing Up	70
References	72
Appendix.....	75
A1 Interview Guide.....	75
A2 A-Categories	78
A3 B-Categories.....	119

1 Introduction

According to studies in the software industry only 34% of IT projects are considered a success (Schwartz, 2004) and the participation of users and customers in software development has become one of the scapegoats. This thesis focuses on stakeholder collaboration in Agile software development, narrowing in on developer and customer collaboration in the development process. Agile methods are a set of development processes, with a high requirement for customer participation, overall concerned with creating and responding to change (J. Highsmith & Cockburn, 2001).

The topic of incremental vs. iterative development has been an on-going discussion since the early days of software development (Larman & Basili, 2003). A favourable approach was for many years Royce (1970)'s paper on the incremental Waterfall Method. However, in the 1990s the iterative community started to blossom with immerging methodologies such as Scrum, Extreme Programming and Feature-Driven Development. In 2001 a group of 17 influential individuals within these fields met in Utah. They coined the umbrella-term *Agile* and wrote the Agile Manifesto, a unified mind-set to iterative software development (Larman & Basili, 2003).

In today's age of rapid changing technology Agile development is more relevant than ever. A good example is the music streaming service Spotify. The company has competed against giants like Apple, Google and Amazon (Sutherland, 2014b), but has reached over 10 million paying subscribers (Brustein, 2014). Jeff Sutherland, one of the co-founders of Agile (Beck et al., 2001) states: *"Google, Amazon, and Apple could crush Spotify in a nanosecond if the company wasn't perpetually striving to be faster, better, and cheaper. To survive, Spotify has to be Agile."* (Sutherland, 2014b). Spotify has had a very systematic approach to Agile, according to Sutherland (2014b). Projects like Spotify, who use an Agile approach, get an edge by delivering recurring value to customers. They avoid extensive planning through the classical Gantt charts because studies have shown that humans

lack fundamental abilities in estimating the time required for activities in a project (Sutherland, 2014a).

By using Agile processes Spotify attain recurring input from their users and customers. Apple, on the other hand has often been viewed as a company with lack of customer collaboration, Steve Jobs once said: *“It's really hard to design products by focus groups. A lot of times, people don't know what they want until you show it to them.”* (Sager, 1998). However, it has been said that after the Steve Jobs era Apple has started to listen more to their customers (Rowinski, 2014).

It is clear from the above examples, as well as literature (Dybå & Dingsøyr, 2008; J. Highsmith & Cockburn, 2001), that customer collaboration is a high priority within Agile methods. However, several studies have reported challenges regarding implementing customer collaboration in practice. Hoda, Noble, and Marshall (2011) for one have stated that lack of customer involvement is one of the biggest challenges faced by today's Agile teams. The purpose of this thesis is therefore to dive deeper into the collaboration between developers and customers in Agile software development.

1.1 Research Questions

The research done in this thesis is grounded in a paper concerning the impact of inadequate customer collaboration by Hoda et al. (2011). The paper used a Grounded Theory approach to investigate the lack of customer involvement in Agile projects. This thesis will further advance and qualify the findings from Hoda et al. (2011) by changing the context. Hoda et al. (2011) took a wide approach by interviewing 30 practitioners, in 16 different development organizations in New Zealand and India. This thesis will take a more narrow approach by focusing only on one development project. This is the same approach that Kautz (2011) took when studying user and customer collaboration in the German software project “Waterworks”. Kautz (2011) held semi-structured and open-ended interviews with the development team of

“Waterworks”. The research done through this thesis will have a similar approach. The thesis will study the projects “*Rovbase 3.0*” and “*Feltdagbok*”, where the consultant company Bouvet have been contracted by a governmental organization. The project concerns the development of database applications for collecting data about nature and wildlife in Scandinavia. The thesis will cover the developer’s collaboration with the customer of these systems, rather than the user.

The research from Hoda et al. (2011) focused on finding the causes, consequences and contingencies regarding the lack of customer involvement in Agile software projects. This thesis will focus on the same elements and hence deals with three main areas. These three areas are written as three research questions, which will now be presented.

1. Cause: What are the reasons that developers experience problems when collaborating with customers?

A project involving developers and customers will inevitably lead to some challenges in collaboration. What are the causes for these challenges? The point of this research question is to explore what barriers exist in the collaboration between developers and customers. Typical barriers can be:

- *The distance factor* between the developer and the customer
- *The lack of time commitment* from the customer
- The customers show *scepticism or hype* towards Agile methods

Proposition: The case project has challenges with customer collaboration due to the *lack of time commitment* from the customer.

2. Consequence: How do problems with customer collaboration affect the development team?

The challenges in collaboration between the developer and the customer will inevitably lead to some ramifications. What exactly are these ramifications? This research question is created to shine light on the issues a software project

experiences when collaboration problems exists between developers and customers. Typical consequences are:

- *Problems in gathering and prioritizing requirements*
- *Problems in securing feedback* from customers
- *Loss in productivity* and even a *business loss*.

Proposition: Developers especially experience *problems in securing feedback* because customer representatives are decentralized.

3. Contingency: What strategies do developers use to deal with challenges regarding customer collaboration?

The causes of collaboration challenges and the ensuing consequences drive developers to carry out strategies for dealing with the challenges. What are these strategies? The last research question will explore the ways developers mitigate the consequences of customer collaboration. Typical strategies are:

- *Providing options* to the customer
- Assign a *customer proxy* to especially find and voice the opinions of the customer
- Using *e-collaboration* to mitigate the challenges that arise when the developers and customers are decentralized

Proposition: The case likely uses *e-collaboration*, such as video conferencing and online demos in dealing with customers in different countries.

1.2 Context

The research questions are made to clearly define *what* will be researched. Another important precondition to establish is *why* this research is being conducted. Since the 90s, Agile methods have increasingly become a trend for managing development projects within the software industry. The emergence of Agile methods, such as Scrum, is largely due to the desire from developers to increase the value delivered to customers (Sutherland, 2014a). However,

this also sets some expectations on the customer. With a project operating in Sprints, the customer is required to give recurring requirements and feedback, ideally in-person, to the development team. This is doomed to lead to some challenges in communication and collaboration between the customer and development team. The research motivation lies in contributing to these industry issues by identifying the root challenges and consequences they have on software projects and finding the strategies that can be used to solve them. The goal is that developers, project leaders, managers and even customer organizations can use this thesis to overcome challenges they are experiencing in their development projects. This may lead to improved productivity, better business results and overall better collaborative environments between developers and customers.

“We must learn what customers really want, not what they say they want or what we think they should want.” (Ries, 2011, p. 55)

1.3 Definitions

In this section the main keywords of this thesis will be defined. Firstly, Agile can be defined as: *“Agile development combines creative teamwork with an intense focus on effectiveness and manoeuvrability”* (J. Highsmith & Cockburn, 2001, p. 120). The four core values of Agile Methods are: (1) Individuals and interactions over processes and tools, (2) working software over comprehensive documentation, (3) customer collaboration over contract negotiation and (4) responding to change over following a plan (Beck et al., 2001; J. Highsmith & Cockburn, 2001). Hence, Agile is not a set of principles, but a mind-set to software development, overall concerned with creating and responding to change (J. Highsmith & Cockburn, 2001).

The difference between *user* and *customer* is not always clear. Some papers have a clear and defined difference (Hope & Amdahl, 2011; Kautz, 2011), while some papers have an unclear definition or use the terms interchangeably (Hoda et al., 2011; A. Martin, Biddle, & Noble, 2010; Subramanyam,

Weisstein, & Krishnan, 2010). Kautz (2011) states that especially within Agile literature the distinction between users and customers are often unclear. The reason for this is potentially that different Agile methods use different terms for user and customer (Jim Highsmith, 2002; Rasmusson, 2010). For example the Agile method Crystal sticks to the term *user*, Scrum uses the terms *Product Owner* and *customer*, while XP uses *customer* and *user* nearly interchangeably (Jim Highsmith, 2002). Despite of this, it is practical to have a definition established, which is uniform for the different Agile methods.

”Customers and clients are the ones who have ordered the information system, who will pay for it, and whose requirements are initially specified” (Kautz, 2011, p. 219). According to A. Martin et al. (2010) the customer is the voice of the client organization. Although some papers have said that this voice is coming from only one person, it is suggested that it is usually a team of people acting as the on-site customer. Their responsibilities are; driving the project, providing requirements and quality control (A. Martin et al., 2010). The focus of this thesis will be on the customer.

“[Users are] those who are directly using the information system and deploying it in operation” (Kautz, 2011, p. 219). As with the customer, the user will have someone voicing their opinion; this can be managers or hired professionals, hence not necessarily the actual end-users (Hope & Amdahl, 2011). Customers can also be users of the IT-system, but the users are not always the customers. Therefore, it is important to separate them into two different stakeholder groups (Hope & Amdahl, 2011; Kautz, 2011). According to Subramanyam et al. (2010) users should participate in the following activities: (1) Project scoping and prioritization of requirements. (2) Responding and providing inputs to product prototypes created by development teams. (3) Participating in design meetings and providing inputs on product features.

It is also interesting to look at differences between the keywords *participation*, *involvement* and *collaboration* in literature. Literature researched in this thesis indicates that there are no distinctions between these terms, and they are used interchangeably (Hoda et al., 2011; Hope & Amdahl,

2011; Kautz, 2011). This thesis will mainly use *collaboration* when talking about the relationship between the developer and the customer. According to J. Highsmith and Cockburn (2001) collaboration and participation is when all players, sponsors, customers and developers are on the same team.

”Participation is not just something that is practiced or not practiced. It is a system for interaction that is created in response to different interests, needs and resources, whose qualities may vary considerably from one another” (Hatling & Sørensen, 1998, p. 173).

1.4 Method

The chosen case company for this thesis is the Norwegian company Bouvet. Bouvet is the developer of the management and database application “Rovbase 3.0” and “*Feltdagbok*”. The project customers are state organizations called the Norwegian Environment Agency and the Swedish Environmental Protection Agency (Agency, 2015b). Even though the project spreads across multiple countries, the development team is located in Bouvet’s offices in Trondheim.

The method of collecting data was chosen to be semi-structured and open-ended interviews with the Bouvet development team working on Rovbase 3.0 and Feltdagbok. Each developer was interviewed in-person, and the interviews were recorded and transcribed. To organize and analyse the interviews, a codification process was used to process and funnel the most essential information. To read more about the methodology of the thesis see chapter 3.

1.5 Outline

The disposition of the thesis is organized in chapters, introduced by a preface and abstract. The first chapter is the introduction, which describes the research focus. The next chapter is the theoretical framework. This chapter

covers the state-of-the-art on Agile methods and collaboration between developers and customers. The third chapter will in detail describe the methodology used to attain relevant data, including a description of the interview process and limitations. The fourth chapter is the results chapter. It presents the research findings of the thesis. After the results, the fifth chapter will cover the research discussion. The discussion presents a discussion between the theoretical framework and the results. The last and sixth chapter focuses on the implications of the thesis and gives recommendations for how this topic could be further advanced in research. There is also an appendix, which includes the interview guide and data from the performed interviews.

2 Theoretical Framework

The theoretical framework presented in this chapter is based on a theoretical review by Torstensen (2014). The paper was a state-of-the-art review focusing on relevant theory regarding Agile methods and the collaboration between developers and customers. The chapter is divided into six sections:

#	Topic	Relevance
2.1	Agile software development	This is the principal theoretical field in the study, and it is therefore important with an independent review of its meaning and origin.
2.2	Types of Agile methods	Agile software development is comprised of several different development methods. It is important to present the most relevant of these methods.
2.3	Collaboration	To get a holistic view of the research topic, it is also valuable to review the literature on developer and customer collaboration.
2.4	Collaboration in Agile methods	Lastly, and bringing the other topics together, is the review of collaboration in Agile methods. This also includes sub-sections on causes, consequences and contingencies of developer and customer collaboration problems.

Table 1: Outline and description of chapters

2.1 Agile Software Development

Agile methods are iterative styles of software development often referred to as modern contradictions to the incremental Waterfall style of software development (Larman & Basili, 2003; Rico, Sayani, & Field, 2008). However, according to Larman and Basili (2003) the discussion of incremental versus iterative software development dates back decades. The Agile research community is well aware of this, but some governmental organizations and

commercial actors are not. According to Larman and Basili (2003) iterative software development came independently from countless unnamed project and contributions, but can be traced back to research as early as the 1930s. The first software project to use an iterative approach was said to be the NASA project Mercury in the early 1960s (Larman & Basili, 2003).

Fast forward one decade and Royce (1970) described what would later be known as the Waterfall model. Figure 1 below shows the four different models that Royce (1970) suggested:

1. Simplest form of Waterfall method. It can only be used for very simple implementations.
2. An extended approach to software development is required for more complex implementations.
3. It is necessary to have a fallback position in case steps need to be re-done.
4. Before the project is complete and delivered, it is suggested to do the steps in the Waterfall method twice.

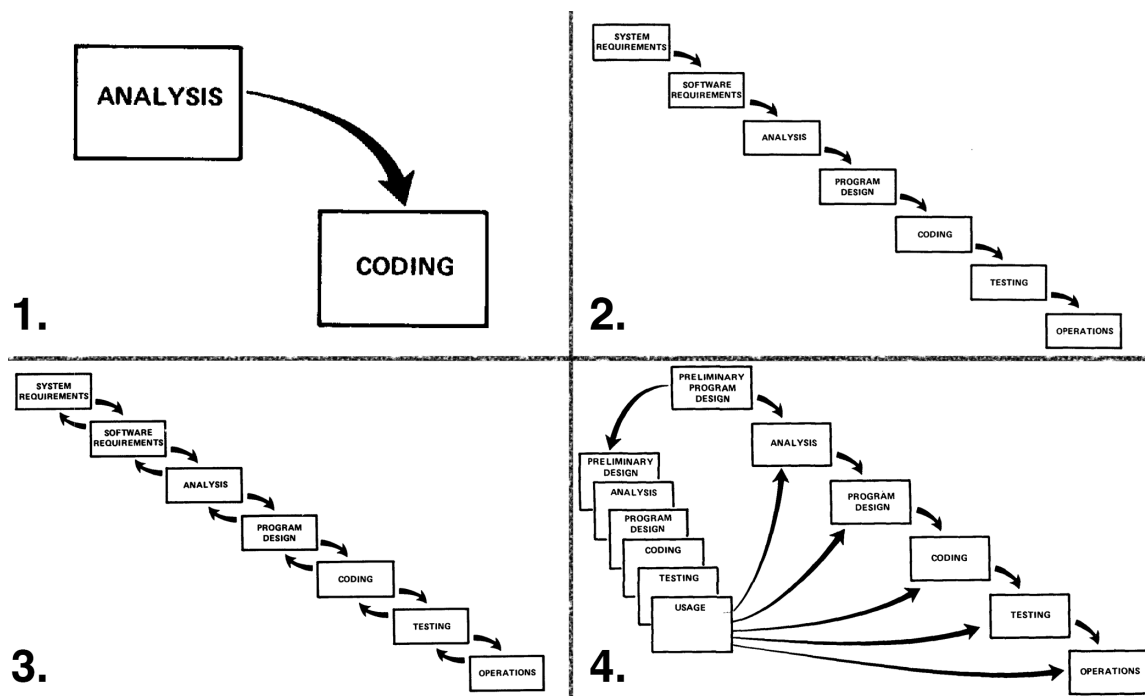


Figure 1: Different models of the Waterfall approach (Royce, 1970)

Figure 1 shows an incremental approach to software development, however the model is not imposing a one-way process. Since the article was published, many have confused the view of Royce (1970) as being that of a single-pass development process, which it is clearly not (Larman & Basili, 2003). Royce (1970) shows a semi-iterative approach and actually stresses the need for customer involvement in the development process. Despite of this confusion the Waterfall method and incremental development has been an important counterweight in the emergence of Agile methods, underlined here: *"Much of present-day software acquisition procedure rests upon the assumption that one can specify a satisfactory system in advance, get bids for its construction, have it built, and install it. I think this assumption is fundamentally wrong, and that many software acquisition problems spring from that fallacy."* (Frederick P. Brooks, 1987, p. 14)

Before the 1990s, iterative development was to a large degree covered by a minority of Waterfall critics. However, in the later half of the 90s the iterative community started to blossom with methodologies such as Scrum, Extreme Programming and Feature-Driven Development being created (Larman & Basili, 2003). With different methods arising over just a few years, the community became very scattered. In February 2001 a group of 17 pioneers and practitioners of the new methodologies met in Utah to find a common ground. The meeting initiated the Agile Alliance, a unified community adopting *Agile* as an umbrella term for the new iterative methods (Beck et al., 2001; Larman & Basili, 2003; Rico et al., 2008).

The Agile Manifesto's highest priority is to; *"satisfy the customer through early and continuous delivery of valuable software"* (Beck et al., 2001). Agile is about stripping away unnecessary amounts of planning and documentation writing, and welcoming more creativity, flexibility and change (Cockburn & Williams, 2003; Dybå & Dingsøy, 2008; Rasmusson, 2010). As mentioned J. Highsmith and Cockburn (2001) state that: *"Agile development combines creative teamwork with an intense focus on effectiveness and manoeuvrability"*. Hence, Agile stresses short development iterations and frequent check-ins with the customer. The four core values of Agile

development are:

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

(Beck et al., 2001; J. Highsmith & Cockburn, 2001)

The list of values is a clear sign that Agile is much closer to being a development philosophy than a set of strict rules. The principles are general guidelines for delivering high-quality software in a recurring manner. The point is not to create a dependency with an inclusive set of rules. However, it is important to follow the principles of Agile. J. Highsmith and Cockburn (2001) state that the team isn't Agile if their customer feedback loop is several months apart. In more recent articles a lack of a clear framework within Agile has been identified: "*Despite the copious research on Agile software development and its ramifications, one cannot help but sense a lack of a unified framework that brings coherence to the seemingly disparate streams of research being pursued*" (Dingsøy, Nerur, Balijepally, & Moe, 2012, p. 1213).

The lack of a unified framework is not the only critique Agile has met. It has been said that Agile development is just "*old wine in new bottles*" and that similar practices can be traced back to the 1960s and 1970s. (Merisalo-Rantanen, Tuunanen, & Rossi, 2005). Also, it has been said that Agile methods might be more suitable for smaller teams, than larger projects (Cohen, Lindvall, & Costa, 2004). Last, but not least, Dybå and Dingsøy (2008) have stated that the evidence for both benefits and limitations of Agile methods are very low. They reason that the field lacks papers both in number and quality. However, Dingsøy et al. (2012) rapport that the field is improving and that both number and quality of papers are increasing. A potential area of improvement is to focus more extensively on studying mature Agile teams, as this has also been identified to be lacking in research (Dybå & Dingsøy, 2008).

2.2 Types of Agile Methods

The first Agile method is said to have been Dynamic Systems Development Method (DSDM), followed by Rational Unified Process and Extreme Programming (Larman & Basili, 2003). Agile Manifesto states that the founders of Agile include: *”Representatives from Extreme Programming, Scrum, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming, and others sympathetic to the need for an alternative to documentation driven, heavyweight software development”* (Beck et al., 2001). Scrum and Extreme Programming (XP) are said to be the most prevalent methods in practice (Dybå & Dingsøy, 2008; Hoda et al., 2011). Extreme Programming has gotten the most attention within Agile research while there has been a call for more research on Scrum (Dybå & Dingsøy, 2008). This thesis will only include a description of Lean, Kanban and Scrum because of its relevance to the participants of this study.

Lean

The Lean software development is an adaptation of Lean Principles into software development. Lean is originally a production system approach from Toyota (Jeffrey, 2004), said to be a big source of inspiration for Agile software development and the Agile Manifesto (Sutherland, 2014a). The method builds on the seven principles of Lean: Eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity in and see the whole (Poppendieck & Poppendieck, 2003).

Kanban

Kanban means signboard in Japanese and was originally part of the Toyotas Lean production system. Originally Kanban was a way to execute Lean thinking in practice, where a physical board with post-it notes was used as signals in the production line. In software development, it has been adapted as a separate development process from Lean software development. The principles of Kanban are: Visualise the workflow, limit work in progress, measure and manage flow, make process policies explicit and improve

collaboratively. In software development the Kanban board (see figure 2) is essential to visualise the workflow, assign work to the different developers, showing the project priorities and highlighting bottlenecks (Ahmad, Markkula, & Oivo, 2013).

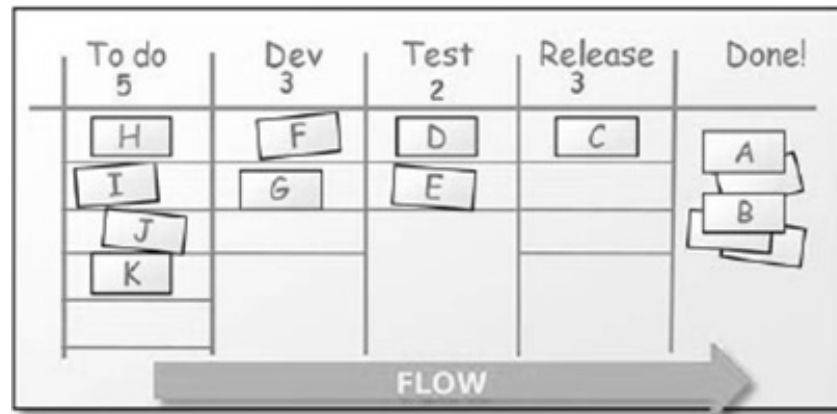


Figure 2: Example of a Kanban board (Kniberg & Skarin, 2010)

Scrum

Scrum focuses on software project management. It is suited for self-organizing teams in situations where it is hard to plan ahead. Software development is done in increments called Sprints; it starts with planning and ends with a review. In charge of voicing the customer is the Product Owner and the team leader is the Scrum Master. First, features that should be implemented are gathered in a Sprint backlog, and then the customers and team decide which features should be prioritized in the coming Sprint. The Scrum Team coordinates the work activities in daily meetings, and the Scrum Master is in charge of identifying problems delaying the team. Finally, the functionality is demonstrated to the customer (Schwaber & Beedle, 2002; Sutherland, 2014a), see figure 3 below.

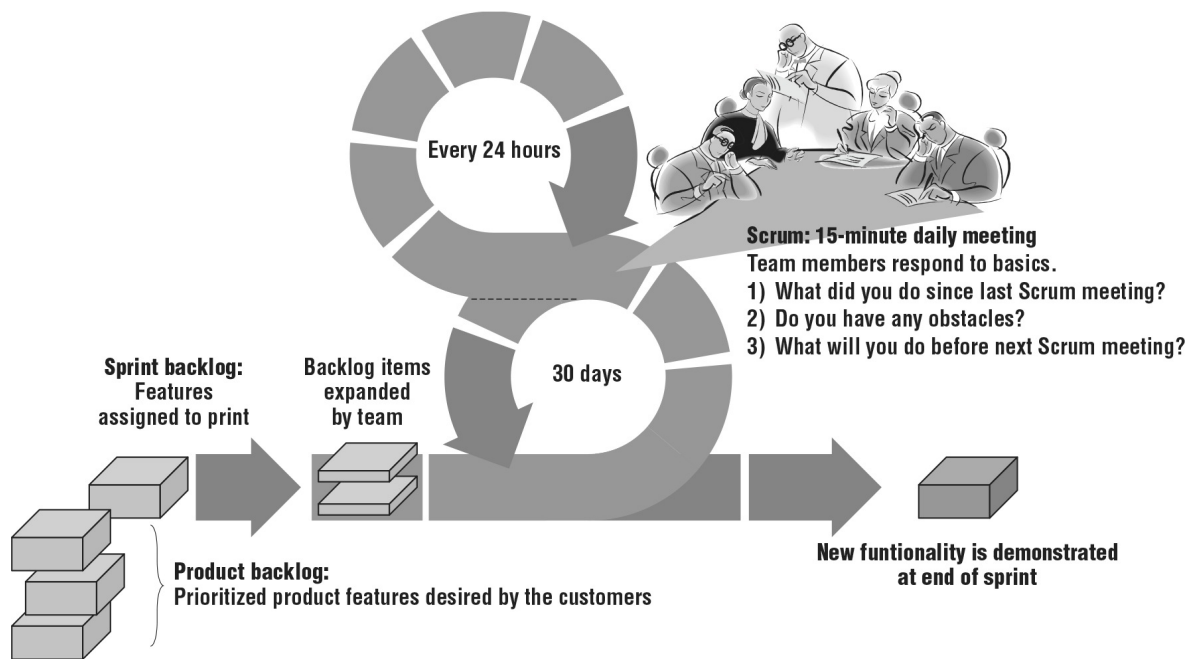


Figure 3: Scrum process flow (Boehm & Turner, 2005)

Now that Lean, Kanban and Scrum have been described the focus will shift over to collaboration. First, a general description of collaboration in software development will be reviewed. After that the focus will shift over to collaboration in Agile software development.

2.3 Collaboration

As mentioned in the introduction previous surveys show that only 34% of IT projects are considered a success (Schwartz, 2004). In fact, user and customer involvement has been reported as the most challenging factor to satisfy (Schwartz, 2004; Subramanyam et al., 2010). However, research has also indicated that developers and users/customers have different perceptions of what *project success* entails. Developers want to develop excellent software and have a clear achievement focus while the users/customers care more about meeting schedules and budgets (Linberg, 1999). Regardless of statistics, or perception of such, it is generally accepted that customers and users of the developed software need to be involved in the process: *"No matter how technically good a piece of software is, it is still poor software if its users*

perceive it as such. Therefore, a level of high user satisfaction is essential to software project success” (Subramanyam et al., 2010, p. 140).

Historically, back in the 70s-80s, computer science theory was more critical to the collaboration between developers and customers in the development process. The acceptance of collaboration has evolved throughout the 90s. The development process has become more and more democratic, and participation of users and customers regarded as an absolute necessity (Hatling & Sørensen, 1998). Collaboration can be seen as an institution external to technology, and the cause-effect relationships between technology and collaboration are an essential area of research (Hatling & Sørensen, 1998).

Studies have indicated that researchers, practitioners and users/customers are all in agreement that collaboration is necessary to achieve a high level of quality and stakeholder satisfaction in IT projects (Hatling & Sørensen, 1998; Subramanyam et al., 2010). One of the challenges is integrating collaboration into the everyday work life of developers and customers. On one hand, software developers often think that they know it best, and hence build boundaries between themselves and the customer (Hatling & Sørensen, 1998). On the other hand, the customer is often resistant to change, indicated in this quote from a developer: *“Yes, I think users are conservative. Users want, to a larger extent, things to remain as they have been, the way they know them from previous experience.”* (Hatling & Sørensen, 1998).

Hatling and Sørensen (1998) said it well when stating that the issue is not whether or not to include user/customer participation in IT-projects, but what role they are given. Hence, it is important to assign clear roles to all stakeholders, especially customers and users, so that they can be effective representatives (Hatling & Sørensen, 1998). Misunderstanding of stakeholder roles happens both in research and practice. Hope and Amdahl (2011) describe that in many of the stages in the Agile development process there is a risk that stakeholders misunderstand their project role. Hope and Amdahl (2011) give examples of projects where user participants were lawyers, more

interested in their disciplines than the system development. Hence, the user/customer representatives are not necessarily voicing the most democratic opinions. This issue is confirmed by this quote from a software developer: “*User representatives seldom have a sufficient overview of the conditions of other users.*” (Hatling & Sørensen, 1998). The role of customers and users are different based on the development approach and also varies from project to project.

2.4 Collaboration in Agile Methods

With the introduction of Agile methods in computer science, the customer is taking on a new role in software development projects. They are no longer in the fringes, but actively shape and guide the evolution of the end product or service (Dingsøyr et al., 2012). The customer is involved through writing user stories, discussing product features, prioritizing the feature lists, and providing rapid feedback to the development team (Hoda et al., 2011). This also sets more expectations on the customer, requiring constant, typically daily, interaction between developer and customer (Jim Highsmith, 2002; A. Martin et al., 2010). One of the core values of the Agile Manifesto states: “*Customer collaboration over contract negotiation*” (Beck et al., 2001). This underlines the importance of customer and developer collaboration in Agile methods projects and underlines that “*Software cannot be ordered like a commodity*” (R. C. Martin, 2003, p. 5). The amount of customer collaboration is different from each project context, but in general, the demand is higher in Agile software projects than in traditional software projects (Hoda et al., 2011).

Theory has suggested several benefits regarding on-going collaboration with customers in Agile methods. Despite the constant demand for onsite customers in the Agile development process it is suggested that: “*[through Agile methods] it is possible to achieve improved job satisfaction, productivity, and increased customer satisfaction*” (Dybå & Dingsøyr, 2008,

p. 20). Studies have indicated that customers are satisfied with the opportunity to give feedback and respond to changes in the Agile development process (Dybå & Dingsøy, 2008; A. Martin et al., 2010). This can also improve the final result because customers are a vital source of information, that developers can leverage (Hope & Amdahl, 2011). A. Martin et al. (2010) suggests that both developers and customers are passionate about their role in an Agile project, enjoy working close together and *"wouldn't do it any other way"*.

In their case study on the German software company AgDev, and their software project for the German public sector, Kautz (2011) identifies an Agile team that found a suitable balance between flexibility and progress. The research was a case study, including semi-structured, open-ended interviews with a third of the development team. The stakeholders considered the project as a success in terms of scope, quality, resources, time, developer satisfaction and user satisfaction. In this project, the customer showed a high degree of satisfaction towards Agile's working methods. A key factor was that the customers and other end users developed trust and a feeling that they had an impact on the development process.

Involvement from users and customers in Agile methods is said to be a sensible strategy, but hard to implement in practice (Hope & Amdahl, 2011). Hence, the theory identifies many disadvantages and issues regarding the close collaboration between developers and customers in Agile software development. First of all studies have shown that the role of on-site customers in Agile methods is unsustainable for long periods of time, making it difficult to introduce into complex and large projects (Dybå & Dingsøy, 2008). It is also challenging to find the right balance of customer collaboration that provides the highest degree of flexibility, progress and quality (Kautz, 2011). The misunderstandings between customers and developers are an important reason to the challenges with customer collaboration in Agile methods. This could be due to a lack of a common frame of reference between customers and developers (Hope & Amdahl, 2011).

Firstly, looking at the view of the customer, they often find it difficult to be

heard in the development process (Hope & Amdahl, 2011). Also, Dybå and Dingsøy (2008) state that the role of the on-site customer can be stressful. In projects that A. Martin et al. (2010) studied the customer had an unsustainable workload, having to handle both the customer role in the Agile project, as well as their own workload. The developers were typically not aware of the long hours the customer was working. This situation is said to pose a great risk for Agile projects. Below are some quotes from interviewed customers regarding their work situation:

- *“I think it’s worked very well, but ... I don’t know how long [I can] keep this pace up”* (A. Martin et al., 2010)
- *“We probably needed about 3 of me, it’s been my life. Look at these grey hairs”* (A. Martin et al., 2010)
- *“I’ve always worked at least 70 - 80 [hrs a week] I don’t even mind it, it’s like what I do”* (A. Martin et al., 2010)

Turning the tables and looking at the view of the developer, Kautz (2011) states that the developers often gets disturbed by the customer constantly interrupting when they are stationed onsite. Also, the developers feel that the requirements provided by the customer is not always understandable (Hope & Amdahl, 2011; Pikkarainen, Haikara, Salo, Abrahamsson, & Still, 2008). The reason could be that the customer lacks understanding of their expectations in Agile projects. Here are some quotes given by interviewed developers underlining the tendency for miscommunication with the customer:

- *“It’s not easy to find out from [customer] what they want; when I say ‘do you want it this way’, they say ‘yes’, and when I ask ‘do you rather want it that way’, they also say yes”* (Kautz, 2011)
- *“Sometimes [customers] only want to come back and see in 6 months what happened [in development].”* (Hoda et al., 2011)
- *“Yesterday he said something and today he says something else”* (Kautz, 2011)
- *“They [customers] are not able at all to formulate their needs in a precise manner. Often, the situation is such that we have to take the task upon ourselves and present solutions to which they may say yes*

or no.” (Hatling & Sørensen, 1998)

A good overview of how developers and users react to different levels of participation in a new development project has been illustrated by Subramanyam et al. (2010). It is illustrated in figure 4 and 5 below. Figure 4 illustrates the level of satisfaction (y-axis) vs., the level of participation (x-axis). The figure indicates that users prefer a low level of participation and become less satisfied when it is increased. Developers, on the other hand prefer a high level of participation, and become displeased when it decreases. Figure 5 illustrates the level of participation vs., the user difference in satisfaction, based on figure 4. This illustrates that users and developers have found a golden mean in a moderate level of collaboration, a level they both can live with.

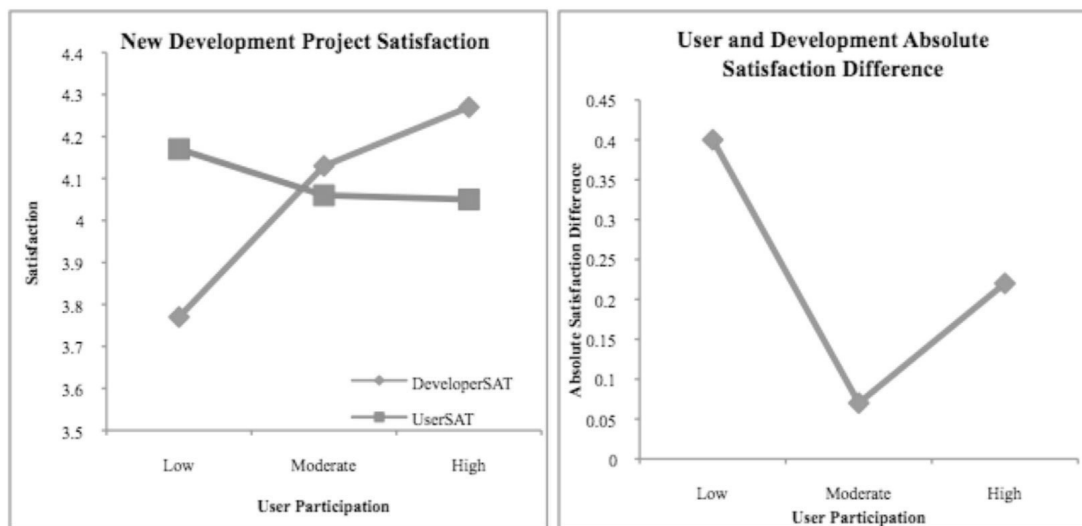


Figure 4 (left) and 5 (right): Overview of developer and user satisfaction, in new development projects, based on user participation (Subramanyam et al., 2010)

Including new development projects, Subramanyam et al. (2010) has also researched how developers and users react to different levels of participation in maintenance projects, as illustrated in figure 6 and 7 below. There are several differences in maintenance projects, compared to new development projects. Firstly, users are less satisfied with low amounts of participation. Secondly, users are most satisfied with middle amounts of participation. Thirdly, both users and developers dislike high amounts of participation. In

general, the developer and users seem to agree more on the level of collaboration in maintenance projects than in new development projects.

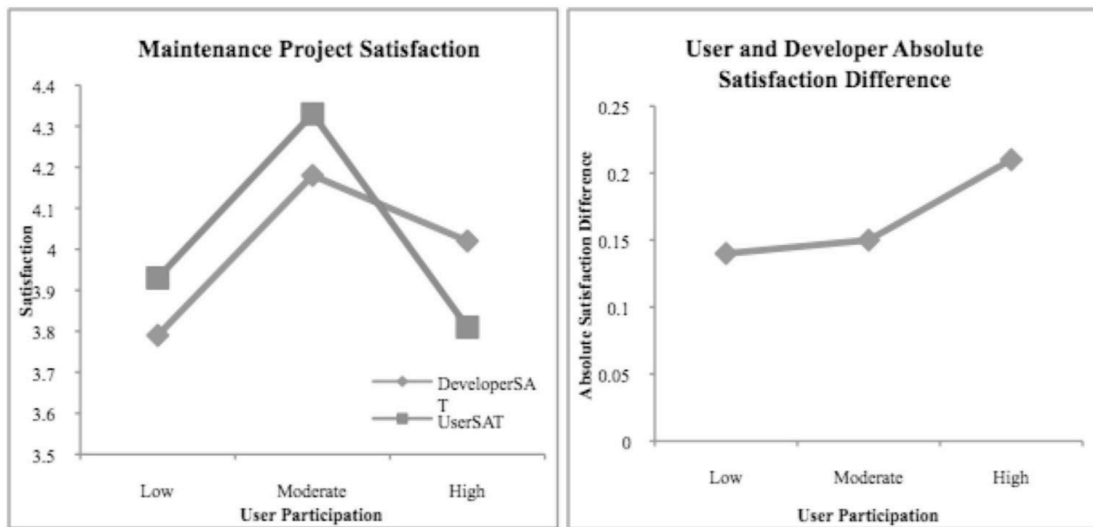


Figure 6 (left) and 7 (right): Overview of developer and user satisfaction, in maintenance projects, based on user participation (Subramanyam et al., 2010)

2.4.1 Cause

Inevitably challenges in collaboration will lead to some problems. This was the research focus in Hoda et al. (2011). The study involved 30 Agile practitioners and 16 software development organizations in India and New Zealand. Hoda et al. (2011) discovered that one of the biggest challenges that Agile teams are faced with is the lack of customer participation in development projects, illustrated in figure 8 below.

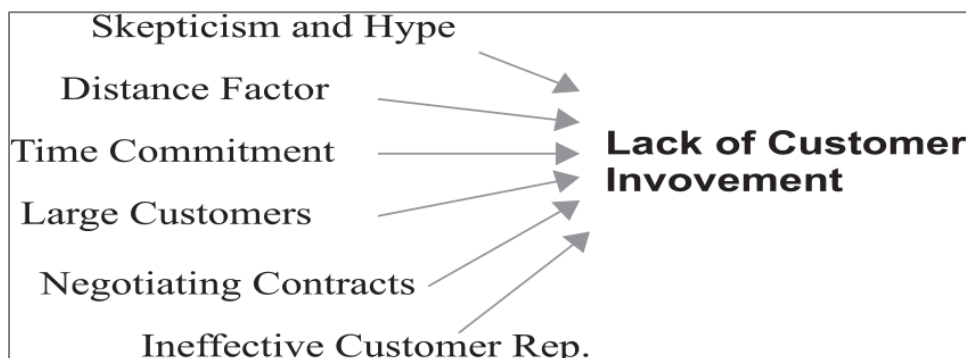


Figure 8: Challenges in customer collaboration (Hoda et al., 2011)

Hoda et al. (2011) described six main causes for collaboration problems between developers and customers, see table 2 below. The causes revolve mainly around the customer’s busy schedule and their willingness to accept Agile software development as a best practice. The customer is in many ways portrayed as someone who is used to traditional development practices, including fixed contracts, documentation requirements and the waterfall process.

Cause	Description
Scepticism and hype	Problems caused by the customer being resistant to move away from traditional ways of software development.
The distance factor	Problems due to the distance between the Agile team and their customer.
Lack of time commitment	A problem source caused by not having enough collaboration time with the customer representative.
Dealing with large customers	Problems due to a large customer or large project, due to less flexibility and openness towards Agile.
Fixed-bid contracts	Problems caused by the customer wanting fixed time, cost and scope, which is a challenge in Agile software development.
Ineffective customer representative	Problems caused by customer representatives being ineffective or lacking in proper understanding of Agile practices.

Table 2: Causes of customer collaboration (Hoda et al., 2011)

2.4.2 Consequence

Hoda et al. (2011) goes on to covering the consequences of collaboration problems between the developer and the customer. There were identified six main consequences, represented in table 3 below. The consequences are mainly due to the limited time from the customers to contribute to the project. The more extreme cases involve the *loss of productivity* and *business loss* as a

consequence. The consequences in table 3 indicate that the risk of unhealthy collaboration in software projects can be severe.

Consequence	Description
Pressure to over-commit	A consequence based on developers under pressure to deliver, especially due to fixed contract constraints.
Problems in gathering and clarifying requirements	Consequences regarding the developers challenge of retrieving and clarifying requirements from the customers.
Problems in prioritizing requirements	The consequence of developers being confused regarding what feature to deliver when.
Problems in securing feedback	Consequence that developers did not get enough feedback from the customers on new features.
Loss of productivity	The consequence of a project experiencing delays and loss of productivity
Business loss	An extreme consequence regarding the loss of business for one of the project companies.

Table 3: Consequences of customer collaboration (Hoda et al., 2011)

2.4.3 Contingency

Lastly, Hoda et al. (2011) presents the strategies that developers use to work around collaboration problems with the customer. There were identified as many as ten strategies used by developers. The strategies were quite different from each other, but revolve around finding ways to satisfy the customer by being flexible in the Agile approach. Most strategies were quite innocent, like trying to change the customer's mind-set towards Agile. *Extreme Undercover* on the other hand, was more on the extreme side. Here the Agile team actually use Agile development practices, but keep the customer unaware, i.e., the developers secretly follow Agile methods behind the customers back. See the complete list of strategies in table 4 below.

Contingency	Description
Changing customers' mind-set	Strategy regarding changing the mind-set of the customer and encouraging them to see the benefits of Agile development.
Providing options	A strategy involving offering alternative options for Agile to get the customer to try the method out.
Buffering	The strategy of adding a buffer time to an estimated time in fixed deadlines cases.
Changing priority	Strategy of adapting the priority of user stories around the schedule of the customer.
Risk Assessment up-Front	A strategy involving pre-project studies to evaluate the risks of an Agile project with a given customer.
Story owners	Strategy regarding splitting up customer representatives into different iterations, so they are not needed for the entire duration of the project.
Customer proxy	A strategy involving assigning a member of the development team to co-ordinate with the customers.
Just demos	The strategy of giving demonstrations of working software to the customer.
E-collaboration	Strategy of using digital communication such as phone, email, chat and voice/video conferencing.
Extreme Undercover	Strategy of using Agile practices internally within the team, but keeping the customer unaware.

Table 4: Contingencies of customer collaboration (Hoda et al., 2011)

The description of Hoda et al. (2011)'s causes, consequences and contingencies is a fitting end to the theory chapter. It is the most essential theory presented because the research questions in this thesis are based on verifying and building upon these findings. These findings, and the findings in the theory chapter in general, underline that the overall challenge with user and customer participation in Agile software development is stakeholder communication. Also, it is easier said than done to require users and

customers to get involved on such a frequent basis. The method, which is the next chapter, will describe the process used in this thesis to gather empirical data. This includes research participants, analytical steps and limitations.

3 Method

The methodology used for the research in this thesis builds around the research questions identified in the introduction:

1. Cause: What are the reasons that developers experience problems when collaborating with customers?
2. Consequence: How do problems with customer collaboration affect the development team?
3. Contingency: What strategies do developers use to deal with challenges regarding customer collaboration?

According to Oates (2005) a research project needs a strategy, a data generation method and a data analysis approach. The research strategy of this thesis was chosen to be a case study, the data analysis qualitative, and the data generation method was chosen to be semi-structured interviews.

This research project's chosen methodology is greatly inspired by the qualitative approach called Grounded Theory created by Glaser and Strauss (1967). GT is an approach where the theory is grounded, i.e., evolved from the conducted field research. This contrasts the deductive approach of first developing a theory and then evaluating it by doing field research (Oates, 2005). The starting approach from this thesis differentiates a little from the classical GT approach in that a theoretical review has been conducted in advanced, and some propositions have been made previous to the field research. This has affected the structure and questions in the interview. However, the mind-set of the field research was to have a wide and open empirical collection process. Hence, the product of the thesis has to some degree evolved as the research progressed.

The GT approach was also used by Hoda et al. (2011). The paper found the approach especially relevant for research regarding collaboration between developers and customers in Agile projects. The three main reasons Hoda et al. (2011) found a fit between GT and Agile projects were:

1. In being a qualitative approach, GT is well suited for studying social interactions and behaviour
2. GT is especially created for young research areas such as Agile, or areas that have not been studied in great detail earlier.
3. GT is an approach that is increasingly being used to study Agile teams.

3.1 Participants

The primary case company in this research is the consultant company Bouvet, and the secondary case companies are the governmental organizations called the Norwegian Environment Agency and the Swedish Environmental Protection Agency. Bouvet ASA, founded in 1995, provides services in the field of information technology, digital communication and enterprise management. They have more than 1000 employees divided among 14 offices in Norway and Sweden (Bouvet, 2015; Brønnøysundregistrene, 2015). The Norwegian Environment Agency was founded 2013 to reduce greenhouse gas emissions, manage Norwegian nature and generally prevent pollution (Agency, 2015a). During the interviews and also in time of writing Bouvet is contracted by the Norwegian Environmental Agency and the Swedish Environmental Protection Agency to develop and maintain the online database applications Rovbase 3.0 and Feltdagbok. Rovbase 3.0 is mainly a maintenance project, while Feltdagbok is closer to a new development project. These projects are the chosen study cases for this thesis.

The work with Rovbase 3.0 started all the way back in 1987, and as the name implies Bouvet is currently developing the third version of this application. Since 1987 Rovbase has evolved to become a management tool and database for carnivores in Norway, including bear, wolf, lynx and golden eagle. It is a crucial tool in the storing and retrieving of information for the public administration and monitoring of these predators. It is a crucial tool in the day-by-day work of numerous Scandinavian county and state officials. (Agency, 2015b, 2015c).

The Rovbase 3.0 development team is also running a second development project with the same customer, called Feltdagbok. The customer for the Feltdagbok project is more specifically the Norwegian Nature Surveillance, a section within the Norwegian Environmental Agency (Surveillance, 2015b). Feltdagbok is a tool for reporting missions and activities in protected areas, as well as violations of the environmental legislation (Surveillance, 2015a).

The research participants of this study were Bouvet employees assigned either to the Rovbase project, the Feltdagbok project or to both. The team working on the two projects are located in Trondheim. They total six people, and include one team leader, one project manager and four developers. The complete list of participants:

ID	Name	Project	Role
#1	Kenneth	Rovbase	Team leader & architect
#2	John Sverre	Rovbase	Developer
#3	Anders	Rovbase/Feltdagbok	Developer
#4	Stian	Feltdagbok	Developer
#5	Anne	Feltdagbok	Project manager & designer
#6	Kay	Rovbase/Feltdagbok	Developer

Table 5: The complete list of participants

There are several reasons for the choice of the Bouvet team as an empirical source. First of all, they were initially interested in the research topic, which is a good start. Secondly, they met two requirements necessary to include them as participants. The first requirement was that they had to use at least one Agile method in their development and the second requirement was that they were actively communicating with the customer in the development process. Thirdly, more research on experienced Agile teams has been called for in previous literature (Dybå & Dingsøy, 2008; Hoda et al., 2011), which the Bouvet team also qualify for.

As an Agile method, the Bouvet team use what they state as a mix between Kanban and Scrum. Their planning cycle is inspired by Scrum, while their task

management is based in Kanban. See image 1 of the Bouvet Kanban board below. Bouvet does not operate with fixed Sprint lengths but push finished software sporadically out to a test server, where the customer can test it.

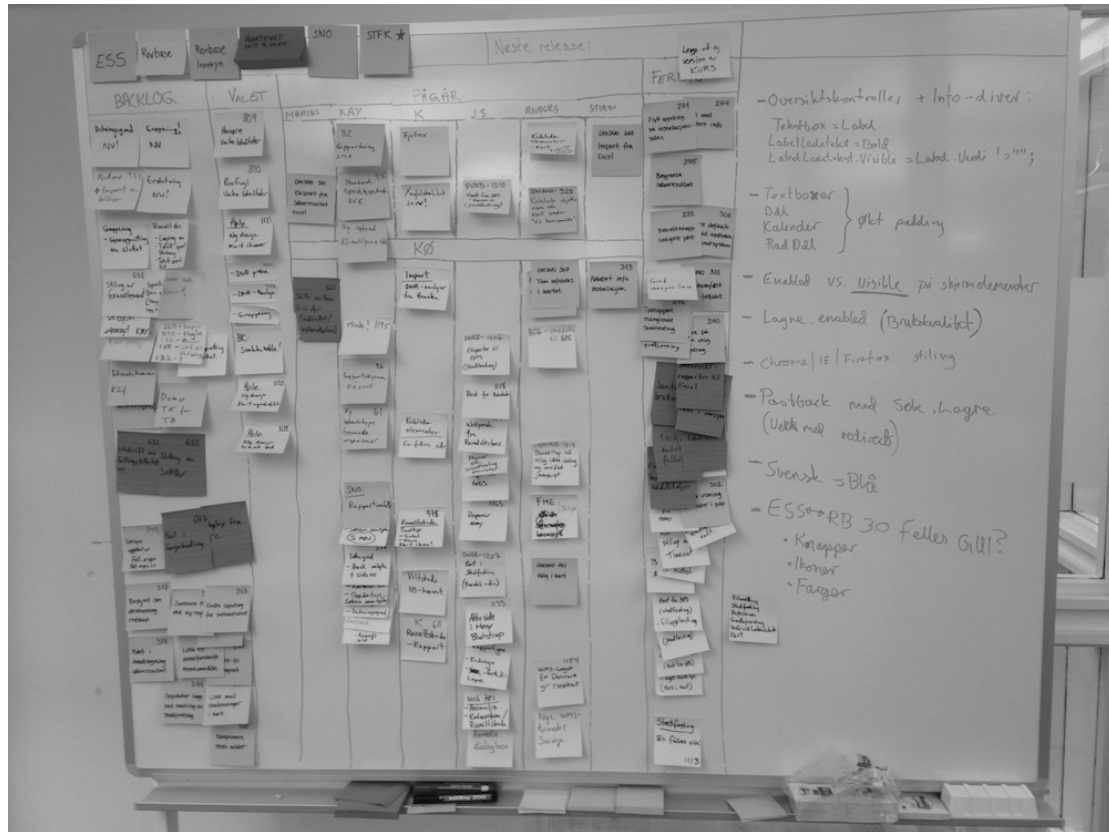


Image 1: Kanban board for Rovbase and Feltdagbok

3.2 Interviews

The interviews with Bouvet have been done in a semi-structural manner. The semi-structural interview typically follows a predefined set of questions, usually with a recurring theme (Tjora, 2012). The questions asked to the participants follow a predetermined order but allows participants to converse freely after each question is set. Where appropriate the participants are also allowed to elaborate further on topics they find important.

The predetermined questions were constructed before the interviews in an interview guide. The interview guide can, in its entirety, be read in the Appendix of this thesis. It is also recommended to practice the prepared

questions with a test subject (Oates, 2005). Therefore, to make the interviews as relevant as possible the interview guide was tested on three independent test subjects. In this process, questions were added, removed or rephrased to improve the overall quality of the final interviews. Including the improvements on the interview guide this gave valuable practice to the interviewer in posing the questions in an understandable and unambiguous manner.

3.3 Analytical Steps

Throughout the gathering of empirical data in this thesis, it was imperative to thoroughly record, transcribe and analyze the results in a stepwise manner. The empirical results were taken from a word-by-word transcription to a higher level of abstraction. This process is essential to generalize the data from interviews and to introduce new models and theory (L. Ø. Widding, 2005). To process the interviews the stepwise process from L. Ø. Widding (2006) was used, see figure 9 below. This process is again inspired by the GT approach in Glaser and Strauss (1967). The figure shows a systematic method for processing empirical data and generating new theoretical suggestions.

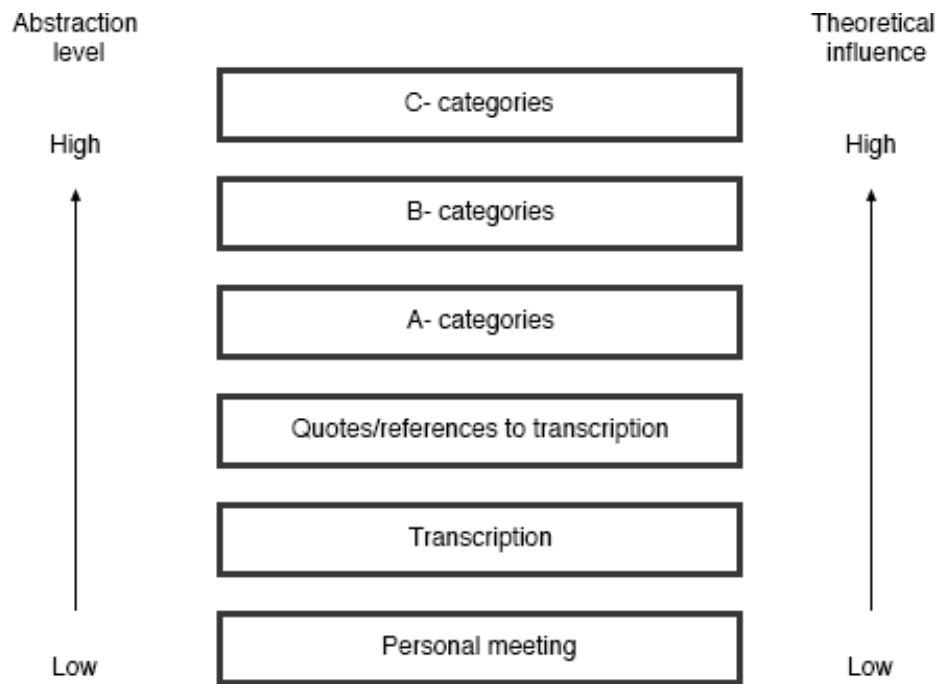


Figure 9: Illustration of the Analytical steps used, based on L. Ø. Widding (2006)

Step 1: Personal meeting

During interviews, it is preferable to meet the interview objects in-person, rather than through other forms of communication (L. Ø. Widding, 2006). Hence, all developer subjects were met in person and interviewed at Bouvet's offices in Trondheim. All interviews were recorded in their entirety using two separate tape recorders.

Step 2: Transcription

It is essential to transcribe each spoken word of the conducted interviews (Oates, 2005; L. Ø. Widding, 2006). Therefore, all audio recordings of the interviews were listened through and transcribed word for word.

Step 3: Quotes/references to transcription

When the transcription is complete, the coding process can start (L. Ø. Widding, 2006). The mind-set of this thesis was to have a fairly open coding process, but that the research questions should mainly guide the chosen quotes and references. This approach is supported by Tjora (2012), who

indicates that the empirical data should be categorized based on an existing theoretical framework.

Step 4: A-categories

The A-categories are the first step in the coding process and is the lowest level of abstraction. This consists of clean quotes, directly from the transcription (L. Ø. Widding, 2006). The quotes from the participants were given a code based on interview number, quote number and to which research question it was relevant. An example of an A-category code used is A-1-001-1, see figure 11 below for more information. If a quote had no relevance to any of the research question but was still found relevant a “o” was used. To see the complete set of chosen A-categories see the Appendix. To qualify the chosen quotes to the research questions, an independent third person was asked to confirm the relevance of the A-categories.

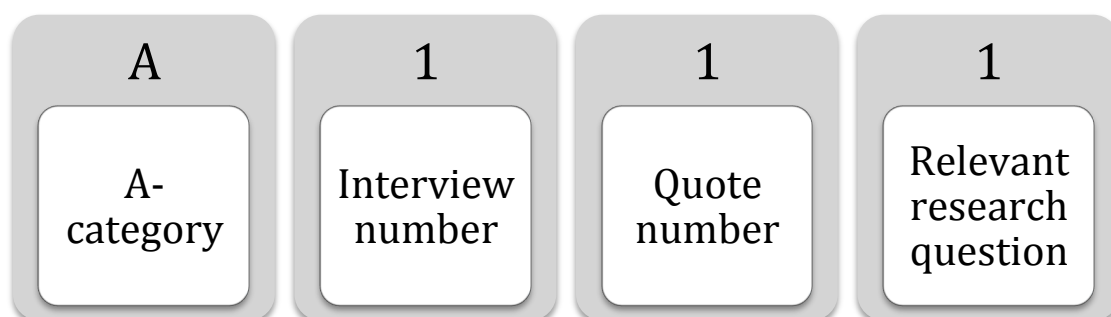


Figure 11: Meaning of the codification A-1-1-1

Step 5: B-categories

In the B-categories, the A-categories are reviewed and compared to the theoretical framework to create a set of more refined empirical data with a higher level of abstraction (L. Ø. Widding, 2006). Here the chosen A-categories were refined mainly based on relevance to the research questions. The chosen set of A-categories were then translated from Norwegian into English. To see the complete set of B-categories see Appendix. The translation and screening of A-categories in this step was also quality checked by an independent third person.

Step 6: C-categories

In the last step, the B-categories are compared to find similarities and differences. In this way, it is possible to give some theoretical contribution (L. Ø. Widding, 2006). The C-categories are represented in the discussion, implication and further research of this thesis. They are the basis for models, discussions, theoretical contributions and suggestions for further research. Hence, the c-categories are the highest level of abstraction and theoretical influence.

3.4 Limitations

There are several potential limitations present in the methodology of this research. Firstly, it is important to highlight the subjective nature of the interview objects. Through the interviews, the participants share personal impressions, i.e., biases, in the state of events. This is also underlined by Hoda et al. (2011). Therefore, some information can be withheld or disclosed, deliberately or unknowingly, which differs from the true occurrences of a given event. One reason for giving inaccurate information can be a misunderstanding from a subject on the an interview question. However, when dealing with communication between individuals, which this thesis does, many incidents are open to interpretations and do not hold the absolute truth. Hence, to evaluate and discuss different events it has been essential to get interpretations from different interview objects regarding the same events.

Secondly, there is a potential limitation when the conducted interviews are transcribed. Firstly, subjective interpretations from the author of this thesis are a potential limitation. Including this, unclear audio may have caused some sentences to be misinterpreted. According to Oates (2005) transcribed notes from an audio interview may not capture non-verbal communication present during the actual interview. In the transcription process sentences that were hard to interpret have been left out from the interview. Also, some notes have been taken during the interviews to capture some of the non-verbal cues from

the interview subjects. Through the different analytical steps, third persons have also been used to validate the transcribed data from the interviews.

Thirdly, it is possible that including only the Rovbase 3.0 and Feltdagbok project could cause homogenous answers. However, Kautz (2011) based his research on only one case project. He states that it is often declared that it is difficult to generalize and theorize on a single case study. However, papers such as Walsham (1995) suggests that one can generalize case study findings in the form of a contribution of insight. Therefore, only two case companies can give valuable insight and build a solid foundation for further research.

Lastly, there are some specific limitations based on the chosen participants. Firstly, there were some developers that had been involved in the project a very short time. Secondly, no agreement on anonymity was signed before the interview. This could cause the developers to hold back on certain information about the customer, which they would not want in a public thesis. Lastly, only the developers were interviewed, not the customers. This might cause the results in the discussion to be mostly in the view of the developer. These participant limitations are however also present in for example Hoda et al. (2011), who also focused mainly on the developer viewpoint. Interviewing multiple developers though, helps to give a more holistic viewpoint in the results.

This concludes the identified limitations, as well as the method chapter. This leads into the next chapter, which is the results chapter. This chapter will present the empirical data extracted through the analytical steps of the method.

4 Results

This chapter covers the research results of the thesis. It covers the most essential findings from the developer interviews. The results will be organized around the three research questions presented in the introduction of the thesis.

1. Cause: What are the reasons that developers experience problems when collaborating with customers?
2. Consequence: How do problems with customer collaboration affect the development team?
3. Contingency: What strategies do developers use to deal with challenges regarding customer collaboration?

Within each of the following sections, the findings from the interviews that were found most relevant to the given research questions will be presented. This is important in order to clearly distinguish between the statements regarding causes, consequences and contingencies. The presented results will be the basis for the discussion in the next chapter.

4.1 Cause

In this section research question 1 will be covered, focusing on the causes of developer and customer collaboration problems. The findings cover what the developer interviewees found problematic in their cooperation with the customer. This is crucial to be able to identify the root of these problems.

The findings for research question 1 are summarized in table 6 below. The table shows how the interviewees view the causes for collaboration problems between them and the customer. It also shows how much this cause influences their collaboration with the customer, i.e., the level of relevance to this project. The grading criteria is as follows:

- High: One or more interviewee(s) view this as a serious problem in their collaboration with the customer

- Medium: One or more interviewee(s) view this as a mediocre problem in their collaboration with the customer
- Low: One or more interviewee(s) view this as a minor problem in their collaboration with the customer

Cause	Level of relevance
Lack of time commitment	High
Governmental issues	High
Fixed-bid contracts	Medium
Ineffective customer representative	Medium
The distance factor	Low
Dealing with large customers	Low
Scepticism and hype	Low

Table 6: Interviewees causes for collaboration problems

Lack of time commitment

The customer's busy schedule and their lack of time devoted to the project were by many of the interviewees defined as the most severe problem in the collaboration. Indicated by these interview quotes:

- *What we struggle the most with is perhaps getting hold of the customer, that they are too busy to answer questions. (A-4-6-1)*
- *The only thing is that they are little available at times. So we are struggling to get clarifications. It's actually the biggest challenge we have. (A-5-6-1)*

The developers describe the customer as practitioners, who aren't especially interested in spending their valuable time in supporting the development of an IT-system. They want to practice their profession, and expect the software developers to take care of the development without bothering them:

- *They are practitioners right, they are interested in their field, and then they have an IT-system to help them. They aren't interested to... They just want it to work. (A-6-13-1)*

Governmental issues

Due to the customers in this thesis being governmental, some issues occur which are unique for these customers. Due to sudden changes in governmental regulations and guidelines they often need to change their mind last minute. Hence, due to by bureaucratic processes, they do not appear as the most proactive customer:

- *The customer can all of a sudden introduce new guidelines from the state and government, which must be complied with (A-1-46-1).*

What makes the Rovbase 3.0 project even more unique, is that governments from multiple countries are involved:

- *Some challenges naturally occur when we are working with two different countries. (...) Sweden is subject to many regulations because of their membership in the EU, which we do not have in Norway. (A-1-12-1)*

Fixed-bid contracts

In having the same customer, the Feltdagbok and Rovbase 3.0 project naturally have a lot in common. However, some differences in collaboration issues occur. This is especially on issues regarding negotiations of contracts. Rovbase 3.0, on one side, is a maintenance project. They do not require fixed estimates on every feature, and Bouvet is given a budget every year, that they can allocate the way they see fit:

- *In Rovbase being a maintenance project, then it's not the same limitations. The purchasing process, where they want a price on the product, falls away. With us, we win a contract over a three year period, which includes maintenance of the system. (A-2-13-1)*

In the Feltdagbok project, however, there is much more focus on estimates. This project leans more towards a new development project, then a maintenance project. This means that the customer is more demanding regarding time schedules and pricing for new features:

- *Since it's a governmental customer, they have a budget to consider. So they would like to have a number, an estimate. It can be a bit*

challenging, because in an Agile world, then perhaps we aren't supposed to give an estimate on large parts of the requirements specification. (A-5-7-1)

Ineffective customer representative

Even though it is not stated as a severe problem, there are some customer representatives that are viewed by the interviewed developers as ineffective. This is mostly regarding the customer's lack of understanding and competence regarding work methods of an Agile team and the expectations an Agile project sets on the customer:

- *By some, the understanding is higher, that we can not always give exact figures or evaluations on the job we do. (A-3-11-1)*
- *It is possible, if they knew more about how we worked, that they would be more frequent in giving feedback. (A-4-12-1)*

The distance factor

Bouvet communicates with both customer representatives in the same city, in Norway, and with customer representatives in Sweden. However, the distance factor between the developers and the Swedish customer is not identified as any cause for collaboration problems. This is especially clear from this interview quote:

- *They (Swedish customer), are very good at communicating. They are actually more accessible than the Norwegian customers. Because they, in return, are even more eager on the digital media. (A-2-11-1)*

Dealing with large customers

Even though the developers are dealing with large governmental agencies, they do not express any problems caused by the customer size. The agencies are divided up into sub-divisions, enabling more flexibility and understanding of the Agile processes:

- *Those that the issues do not concern are outside the loop anyway. So even if the customer is large, then maybe those involved are not equally many. (A-2-12-1)*

- *The Norwegian Environment Agency and the Swedish Environmental Protection Agency are large organizations in that respect. But it is my impression that they have a positive attitude towards Agile in spite of their large size. (A-3-9-1)*

Scepticism and hype

The developers are not experiencing any scepticism from the customer regarding Agile methods. Some customer representatives seem very enthusiastic about Agile methods, while others seem indifferent or unaware of what development methods are being used in the first place:

- *I think it is preferable from their (customer) side, (...) they are proud of how it is operated. (A-2-9-1)*
- *I think at least not my customers are 100% aware of how we work. That we are working in a special method. (A-4-7-1)*

4.2 Consequence

The focus of this next section will be on research question 2, where the consequences of developer and customer collaboration problems will be covered. In other words, how do the causes of collaboration influence the workflow, and the results of the development team? The findings cover the consequences identified by the interviewed developers.

The findings for research question 2 are summarized in table 7 below. Each consequence is listed together with their level of relevance. The level of relevance shows to what degree the interviewees view this as an impact for their project. The grading criteria is:

- **High:** One or more interviewee(s) view this as a serious consequence for their project.
- **Medium:** One or more interviewee(s) view this as a mediocre consequence for their project.
- **Low:** One or more interviewee(s) view this as a minor consequence for their project.

Consequence	Level of relevance
Low-value deliveries	High
Problems in securing feedback	High
Problems in gathering and clarifying requirements	Medium
Problems in prioritizing requirements	Medium
Pressure to over-commit	Medium
Loss of productivity	Medium
Business loss	Low

Table 7: Interviewees consequences for collaboration problems

Low-value deliveries

First of all, the interviewees indicated that a severe consequence for the project is delivering a product that does not meet the needs of the customer:

- *It might not be what the customer needs the most, that could be a consequence. That if the customer doesn't actively get involved they will be delivered functionality that may not be what they needed the most at that time. (A-2-17-2)*

This consequence is identified as especially severe in this project due to the nature of the customer. In being a government organization that deals with environmental values it is crucial that the software tools are up-and-running with little or no bugs. In the outermost consequence of an error-prone system could be fatal:

- *Wrongly implemented functionality in the work area of pigeons, predators and carnivores could lead to a issued resolution for imposition of bears, which is simply wrong. Instead of the program showing 20 bears in the area, the real number could be 3, and the resolution shouldn't have been passed. So there are large values on the line, both economical and environmental values. (A-1-39-2)*

Problems in securing feedback

Attaining feedback from the customer on test ready software revisions was identified as another severe consequence by many of the interviewees. When a

new revision of the system is complete, it is pushed to a test environment where the customer should login to test and approve the software. However, the customer will not always respond as fast as the developers would want:

- *We have a routine that when we finish something we add a test. Then we tell the customer that everything on the test is new, and that he should go in and approve that everything is working. And this is often where things stop up a little, it can take time, before he gets started. (A-4-18-2)*

However, some developers indicate that the consequence is not so severe. This is because it is the customer themselves that suffers from the delay of feedback. They suffer because they are the ones that are not receiving the new updates to their system. Hence, to get the updates they want, they need to provide the developers with feedback:

- *It will negatively influence themselves (customer), that they don't get things published. Because we put things in the test environment, and then wait for them. Then it goes pretty fast, because they understand that if they give feedback, then it will be produced. (A-5-17-2)*

Problems in gathering and clarifying requirements

Mostly, the interviewees indicate that attaining system requirements from the customers and clarifying their meaning is not a severe consequence to the project. However, some developers think the requirements arrive too late and are not specific enough. Also, a more dedicated product owner, which can clarify system requirements from the customer, is missed:

- *It is typical that it (requirements) arrives a bit too late. That we make it, and then they see that it was not the way they pictured it. (A-5-15-2)*
- *We don't really have a dedicated product owner. That is challenging because that's the guy I should be talking to. Now I just talk with lots of practitioners, so I miss having that. (A-6-22-2)*

Problems in prioritizing requirements

Most of the interviewed developers indicate that figuring out what system requirements to prioritize, and what order of sequence they should be done, is working well in the collaboration with the customer. Also, the Norwegian and Swedish customers communicate with each other in regards to which of they're requirements Bouvet should prioritize. However, it is also expressed that at times the customer is indecisive and will jump rapidly between what requirements should be prioritized:

- *It could be that one week something is very urgent, men then something just comes inn from the sideline, and then that is very urgent, and then the thing from last week isn't so urgent anymore. We experience that all the time. It is the last thing that is the most urgent. (A-6-23-2I)*

Pressure to over-commit

In general there is not much pressure on any the developers to over-commit to deadlines. Even though things become more hectic towards certain deadlines. Especially on the Rovbase project, the customers are very flexible in regards to deadlines:

- *It is more or less accepted that we overwrite the deadlines, those deadlines that have been set. My impression on the Norwegian Environment Agency as a customer is that they are not very strict on that particular point. It is fine for them if things take a little longer time. (A-3-12-2)*

However in the Feltdagbok project they require more estimates and budgets from the developers they are also stricter regarding deadlines. This puts some more pressure on committing to these deadlines, indicated here:

- *When we made those estimates there wasn't placed a buffer on the number of hours. It was precisely what we thought, without putting on any buffer, which you often do to be sure. So then it burst, the numbers of hours. Then there was a little pressure on the development. (A-4-15-2)*

Loss of productivity

Many of the developers point to a productivity loss as a consequence for this project. Some of the interviewees state that the productivity is limited because the team always has a long backlog to pick tasks from. However, several developers indicate that even though there are always tasks available, these tasks might not be relevant for progression:

- *Yes, if the customer is not active, then it leads to us sitting on the fence. Because then we have to wait for unnecessary feedback, and then there is more communication that wouldn't need to take place. (...) There are always new tasks to do, but it is not certain that it fits to begin with that right there and then. Hence, this leads to downtime and leads to less productivity for us. For the customer, you could say. (A-2-28-2)*

Including irrelevant tasks, the timing of the task might not be optimal either. The break in flow could also lead to a loss in productivity. A break in flow could happen by juggling too many unclosed tasks. This developer indicates that there is some idle time when moving between tasks:

- *Let's say you are working on something, and you are really into it, and then you have to take a break from it, and start on something new. Then you get an answer from the customer on that task, so you have to go back again. But of course, you don't remember everything, so you have to revise yourself on the problem. (A-6-24-2)*

Business loss

A direct loss of business for Bouvet is expressed as a minor consequence in this project. However, some loss due to collaboration problems between the developers and customers has been expressed. Since Feltdagbok is not a maintenance project some financial costs were inflicted on Bouvet due to a fixed price given to the customer:

- *We (Rovbase) are lucky, since we have a maintenance agreement, with fairly ok conditions, so we invoice everything. Not on the SNO (Feltdagbok) project however, they have given some fixed prices. I*

don't know if the price was too low, or if the customer asked for too much, but they have taken costs at their own expense. (A-6-27-2)

4.3 Contingency

The following section will cover research question 3, which is contingencies of developer and customer collaboration problems. Hence, it will focus on the strategies that developers use to handle and overcome difficulties in their collaboration with the customer. The findings include strategies used and their level of relevance.

The findings in research question 3 are summarized in table 8 below. The left column represents the contingencies listed by the interviewed developers, and the right column represents the level of relevance of that strategy in the case project. The grading criteria is as follow:

- High: One or more interviewee(s) indicate that this strategy is frequently used in the collaboration with the customer.
- Medium: One or more interviewee(s) indicate that this strategy is used occasionally in the collaboration with the customer.
- Low: One or more interviewee(s) indicate that this strategy is rarely or never used in the collaboration with the customer.

Contingencies	Level of relevance
E-collaboration	High
Building trust	High
Self-testing	High
Fill up backlog	High
Changing priority	High
Just demos	High
Buffering	High
Allocate resources to other projects	Medium
Customer proxy	Medium

Story owners	Medium
Changing customers' mind-set	Low
Providing options	Low
Extreme Undercover	Low
Risk assessment up front	Low

Table 8: Interviewees consequences for collaboration problems

E-collaboration

The developers frequently use digital communication like email, telephone, chat and video conferencing. It is also indicated by the developers that there are low barriers and little formalities around the use of digital communication with the customer. Some forms of digital communication are favored over others. For example, chat was less frequently used and email very frequently used. However, it was considered more important to pick up the phone in urgent matters:

- *I tell everyone that is starting the team, that it is important that clarifications that are urgent are not taken over email, but should taken over the phone. (A-1-40-3)*

Building trust

A long-term focus from Bouvet is establishing trust with the customer. They build trust by showing honesty, openness and commitment. Bouvet have had the Norwegian Environment Agency and the Swedish Environmental Protection Agency as customers for several years, and the trust level has increased over time. This has given the developers more freedom to influence the design and make decisions without involving the customer:

- *It is a little special with this project, I would say, that the customer has given us so much responsibility to design the system. We are free to take quite a lot of decisions, and in many cases exclude the customer in such cases where it might ordinarily have been more natural to consult with them. (A-2-45-3)*

Self-testing

The developers at Bouvet also leverage the trust that the customers give them. One of the ways they do this is by self-testing. If the customers don't have time to give feedback, the developers test the functionality themselves. In that way, the newest updates won't always have the functionality or timing that the customer wishes. However, according to this quote, the customers quickly realize that they need to participate to get what they want:

- *If they don't come with the necessary clarifications, they won't get the functionality delivered. Then we just keep going, and they learn very fast, that if they want things delivered, they have to deliver themselves. (A-1-26-3)*

Fill up backlog

To avoid idle time when the customers are busy the developers fill up the project backlog with excess work tasks. In that way, they always have a task to fall back on, in case progress with ongoing tasks is dependent on customer feedback. In that way the developers can assure a constant workflow:

- *We always have tasks in reserve, so we ensure productivity, even though the customer may be unavailable. We always have something to do, and we are able to deliver those work hours we have committed to deliver. (A-2-44-3)*

Changing priority

Including having access tasks in the backlog the developers also prioritize tasks according to what fits the customer's schedule. They are flexible with rapidly changing focus from one task to another if it is needed. The Norwegian and Swedish customers also understand that the priority occasionally needs a rapid reprioritization:

- *When something comes up, that is more important than anything else, even though we are right in the middle of another case, it is normal that it gets priority over the other. (A-3-22-3)*
- *If there is a change for Norway (customer), then Sweden (customer) will of course understand that of course the resources have to be allocated there, and visa versa. (A-1-48-3)*

Just demos

A key strategy to involve the customers is of course demos of working software. Bouvet uses three main types of demos: Remote testing, live demo and workshops. In remote testing, the software is basically pushed to the staging environment where the customer logs in and tests by himself. In case some bigger features have been made, a live demo is necessary. In case a longer session of testing is necessary, a workshop is also possible.

Interviewees indicated that both live demos and workshop were rarely used, while remote testing was the standard:

- *We perform the task, and then we push it out on the test server. That is a form of demo I guess. However, we don't sit there and guide them through it. It is up to them to login and click through what we have done. (...) It is rare that we meet and run a proper demo, but it happens. If there are bigger things, then we may have to visit their offices and run a demo. But that is more the exception. (A-6-31-3)*

Buffering

The interviewed developers indicate that estimating delivery times is a challenging task and that putting an extra buffer on the estimated completion date is an important practice. It was indicated as more important to the Feltdagbok project than with the Rovbase project:

- *We try to do that every time we estimate. It's always like that when you estimate, that it takes longer than you think. So you have to put on a buffer, if not it (estimates) will burst. (A-4-29-3)*

Allocate resources to other projects

Bouvet has the opportunity to allocate developers between projects in case the customer is unresponsive. Many of the interviewed developers stated that they had worked both with Rovbase and Feltdagbok. So if, for instance, Rovbase is stuck waiting for a busy customer, developers from that team can support the Feltdagbok team:

- *We flex between different projects here. So we can always re-allocate if there is any idle time in a project. We've been involved in several*

projects within our team, so then we just move our resources between those projects. (A-2-29-3)

Customer proxy

The development team has a customer manager that is in contact with the customer regarding requirements gathering, prioritization of tasks throughout the development and testing. However, the customer manager is not solely responsible for the contact with the customer. He also has other responsibilities such as development and project management. The developers also handle much of the contact with the customer themselves:

- *Kenneth has to a large degree contact with the customer. He participates in specifying the requirements and specification. Then he redistributes the tasks to us. Afterward though, we often take contact directly with the customer for clarifications and for further work. (A-2-42-3)*

Story owners

Having story owners, i.e., a customer contact that is responsible for a specific user story, has occasionally been done in the project. However, it was not a standard procedure. In this instance the developers use different practitioners depending on what predator they are focusing on in the development:

- *We are working with functionality for foxes at the moment. This is a section that the Norwegian Environment Agency and the Swedish Environmental Protection Agency are running. Then they have included stakeholders who can participate. When this is complete we are going into functionality for wolverines, and that's when other groups will be included. (A-1-53-3)*

Changing customers' mind-set

There has been little or no need to change the mind-set of the customer regarding the advantages of Agile software development. The developers have explained to the customers what responsibilities they have when involved in an Agile project. Several developers state that the customers are both enthusiastic and understanding of their responsibilities:

- *Since the start of this project, where I come into it, the method we use has been really incorporated. It seems like there is very good understanding and acceptance for it with the customer. (A-3-19-3)*

Providing options

If the customers were very sceptic to Agile, the developers could have offered options to get the customers to accept it as a best practice. However, as mentioned, the customer has been very open to Agile as a framework. It was stated by the developers though, that they structure their work methods to best fit the needs of the customer:

- *If they have any wishes we restructure so that we address them. We have shaped it so that it is very effective in regards to delivering as fast as possible to them. (A-2-34-3)*

Extreme Undercover

Also, if the customer was sceptic to Agile, a possibility is to keep the work methodology a secret from the customer. Since the customer is both positive to Agile and trusting towards Bouvet, keeping the Agile process undercover was not mentioned as a strategy from the developers. However, it was indicated that some customer contacts are not aware of what development method is being used by the development team:

- *We are always open about using Agile. (A-1-51-3)*
- *I don't think they are aware that we are working in a Scrum/Kanban way. (A-4-28-3)*

Risk assessment up front

Conducting an assessment with the customers before the project to evaluate the risk of using Agile software development was not done by the development team at Bouvet. Hence, it is not relevant as a strategy in this case project.

5 Discussion

In this chapter, the research discussion will be presented. Similar as with the results, the discussion is organized around the three research questions regarding causes, consequences and contingencies of developer and customer collaboration problems. The discussion uses the results in the previous chapter as a basis and compares it to previous research, which was presented in chapter 2. In this way, similarities and differences can be drawn between the empirical data and the state-of-the-art. The discussion is the basis for suggested implications and further research, covered in the next chapter.

5.1 Cause

In general the interviewed developers at Bouvet identify only a small number of causes for problems in their collaboration with the customer. However, there are some issues that noticeably affects the developers attempt to work in an Agile manner. Similar as with Hoda et al. (2011), the issues are connected to the lack of customer involvement, with the *lack of time commitment* as one of the more significant issues. However, Hoda et al. (2011) also describes problems such as *the distance factor*, which was pointed out as minor issue by the interviewed developers. In general, the interviewees give the impression that they and the customer have found a good balance regarding the participation level, similar to Kautz (2011)'s case company AgDev. Both the interviewed developers from Bouvet and AgDev identify trust as a key factor in finding this balance.

It was *the lack of time commitment* by the customer that was most frequently mentioned as a cause for collaboration problems by the interviewees. This is not surprising considering the expectations Agile software development sets on the customer. The interviewed developers view the customers in the Norwegian Environment Agency and the Swedish Environmental Protection Agency as researchers and practitioners. Their focus and responsibilities are the environment and wildlife, not on contributing to the development of an

IT-system. Hence, the customers want the developers to take care of the development, so that they can focus on their own work. This is similar to the case company from Hope and Amdahl (2011), where the customer had little time to devote to the development, due to their responsibilities as practicing lawyers.

The case customer's busy schedule and *lack of time commitment* is well in line with Subramanyam et al. (2010)'s figures on satisfaction vs., user participation (see figures 4, 5, 6 and 7). While developers are generally satisfied with high amount of customer participation, the customer on the other hand prefers lower amounts of their participation in development. The developers at Bouvet pointed out that the participation from the customers at Rovbase and Feltdagbok was actually lower than what they had experienced in other project.

The developers show an understanding of the busy schedule of the customer. However, in a couple of the interviews *ineffective customer representatives* was mentioned, especially the customer's effectiveness and competence was questioned. The developers comment that the customer might be more proactive and responsive if they had more understanding of Agile work methods. According to A. Martin et al. (2010) many customer representatives are extremely busy, and the developers are not aware of their heavy workload. On the contrary, however, the developers at Bouvet generally seem very solution oriented and understanding of their customer's workload.

There was also identified some problems in *dealing with large customers*, and the Norwegian and the Swedish customer can both be categorized as large organizations. However, they do not resemble a typical large customer. A stereotype of a large customer is someone resistant and sceptical to change, indicated by Hatling and Sørensen (1998). Large organizations are also identified by Hoda et al. (2011) as an especially challenging customer. Quite the contrary though, the case customer seems very flexible and open for change. They are embrative of the Agile framework and show trust in Bouvet to deliver what they need. They are also on board in regards to dropping *fixed-*

bid contracts, which is mentioned in the Agile Manifesto as essential. It is crucial, though, despite the trust from the customer, that the developers keep involving them in the development process.

It is especially relevant to look at the differences between the new development project Feltdagbok and the maintenance project Rovbase in the light of Subramanyam et al. (2010)'s figures on satisfaction vs., user participation (see figures 4, 5, 6 and 7). According to Subramanyam et al. (2010) both developers and customers are less satisfied when there are high amounts of participation involved in maintenance projects, in comparison to new development projects where high amounts of participation is more accepted. An explanation can be that maintenance projects are long-term, requiring the participation to be more spread out over time. This is also in line with Feltdagbok, which is closer to being a new development project, and Rovbase, which is closer to being a maintenance project. The interviewed developers from Feltdagbok indicate that they are highly dependent on frequent customer feedback to move forward. On the Rovbase project, however, they indicate that they can go longer periods without the customer's involvement.

Unique for this case project is the character of the customer. What stands out is that the customer is governmental. Actually, multiple governments are involved. This introduces some interesting *governmental issues*. Government organizations operate after laws and regulations that are subject to major and rapid changes. These changes will also effect the requirements of the IT-system. Hence, the developers are expected to be able to change requirements very rapidly in case of such changes. Especially when dealing with different governments, this has the potential to cause collaboration problems. For example, if the two governments change laws around the same time, they might both require Bouvet to change the system after their new requirements and potentially with a short deadline. Then it is important for Bouvet and the governments to communicate regarding what should come first. Hoda et al. (2011) did not include this problem, most likely because of its uniqueness. It is

important though, to also include more unique problems in the discussion of developer and customer collaboration.

Summing up, even though the collaboration between Bouvet and their customer seems to be going smoothly, some problems were identified. It was mainly *governmental issues* and *lack of time commitment* that were identified as a scapegoat for the causes of collaboration problems. The customer is a large one, but have little resistance to change, which is untypical in large organizations.

5.2 Consequence

Similarly and consistent with the discussion on causes of collaboration problems, the empirical data indicate that there are few severe consequences of collaboration problems in the case project. However, most of the consequences found in Hoda et al. (2011), such as *problems securing feedback* and *pressure to over-commit*, were also present to some degree in the collaboration between Bouvet and the customer. The high expectations the interviewed developers put on the customer regarding involvement, mixed with the customer's busy schedule, will naturally bring on consequences. This will be further explored in this section.

Several of the consequences are directly connected to the day-by-day communication between Bouvet and the customer. These include *problems in gathering and clarifying requirements*, *problems in securing feedback* and *problems in prioritizing requirements*. As stated by Hope and Amdahl (2011) the customer is a vital source of information for the project. There needs to be a continuous flow of information from the customer to the developers throughout the project. This includes setting requirements at the start of the project, prioritizing throughout the development cycle and attaining feedback on completed features. It is important that feedback and information from the customer is understandable and that it isn't extensively delayed. The

developers at Bouvet indicate that they are experiencing issues both regarding timing and understanding in their communication with the customer.

The experiences that the developers from Bouvet are having with the customer is very similar to the developers experience with customers in Hatling and Sørensen (1998), Hoda et al. (2011) and Kautz (2011). The customers are considered understanding, interested and trusting by the development team at Bouvet. However, it is indicated that the customers are at times slow and late in their communication. When they first do communicate it is often last minute and unclear. Their attention span is limited, and they are somewhat unfocused regarding their participation in the development process. However, the developers are very understanding of the reason behind these negative customer traits. The customer is extremely busy, and their superiors have allocated very limited time for them to follow up the developers at Bouvet. This means that in some instances the customer is using his spare time to give feedback to the development team. This shows that the customer organizations potentially need to think about allocating more time for their representatives to work on Agile software projects.

The communication problems that the developers at Bouvet are experiencing with the customer may have something to do with how they organize and execute their Agile development. As mentioned earlier the development team do not have fixed Sprint periods. With a fixed Sprint period, they would have a set date and time, for example, every month or every other week, for a product demonstration. Instead of having a scheduled time at the end of a fixed Sprint period, the Bouvet development team upload finished software to their test server and notify the customer that it is time to log-in and test the new feature. This means the customer tests the feature when it suits them. This gives the customer flexibility, but it is also easy to postpone the task. Hence, if fixed Sprint periods were introduced, the customer would have already allocated the time for the product demonstration and testing.

The discussion of communication between the customer and developer is a good bridge to another consequence, namely *low-value deliveries* of software.

It was stated by several of the interviewed developers that it is the customers themselves that suffer if they don't give sufficient information in the beginning, throughout and after the development of new features. This consequence is especially important in Agile software development because delivering valuable software is stated as a top priority in the Agile Manifesto. Despite of the high priority in the Agile Manifesto, Hoda et al. (2011) did not include deliveries of low-value software as a separate consequence. However, both the research done in this thesis and core Agile theory suggest that this is an important consequence.

It is especially important to be aware of *low-value deliveries* of software in projects where the delivery of error-prone software can have hazardous consequences. This is especially companies dealing with humans and nature where the cost of a mistake can be fatal. In the interviews, it was indicated by the developers that in dealing with the Norwegian Environment Agency and the Swedish Environmental Protection Agency there was no room for errors and bugs in the system. Therefore, it is even more important that the customer gets further involved in the development of the system. It's the customer that possesses all the knowledge the developers need to implement the system correctly. This underlines the importance of the customer allocating enough time, and scheduling that time routinely, to make sure it's prioritized.

As discussed, earlier collaboration over fixed contracts, is another important principle and value in the Agile Manifesto. This is especially important when developers are being *pressured to over-commit* due to contract-based deadlines. Sutherland (2014a) is amongst the Agile practitioners who state that people are imprecise when planning long-term. Giving the customers fixed deadlines far into the future is risky. This was no problem for the Rovbase project who have flexible deadlines. However, this was a problem for the Feltdagbok project. Bouvet gave an estimate of the numbers of hours on a set of features, and the developers ended up pressured to finish at the agreed upon time.

Giving the customer an estimate of work hours not only gave the developers pressure to finish within the deadline, it also gave Bouvet a *business loss*. This is because an occurrence took place where the developers gave a fixed price to a customer, which was not realistic. Bouvet ended up having to cover some of the extra work hours out of their own pocket. The discussion of plans, deadlines, budgets and contracts in Agile projects is challenging. If the customer wants more forecasts in their project, like Feltdagbok, it is not easy to just decline that request. Again, it is important to make the customer organization aware of what facilitation is required to carry out a successful Agile project. This is in the customers best interest because trying to carry out an Agile project with long-term Gantt diagrams, and contract based development, is risky business. It can cause re-work and frustration in the communication between the developers and the customers.

Including *business loss*, the interviewed developers indicate that there has also been a *loss of productivity* due to collaboration problems with the customer. Due to communication issues the developers sometimes need to jump too frequently between tasks, which breaks the flow of work. In general, however, the developers express that both the loss of business and productivity is limited. The interviews with the developers indicate that the relationship with the customer is both trusting and productive. Like with AgDev, the case company in Kautz (2011), Bouvet and their customer seem to have found a good balance between flexibility and progress. Also, both parties seem satisfied and even enthusiastic about working together and in an Agile manner.

To sum up, the identified collaboration problems in the case project will inevitably lead to some consequences. Because of a busy customer, these consequences are rooted in attaining information from the customer throughout the development cycle. The reason behind this could very well be the lack of fixed Sprint periods. The developer also expresses a consequence in *low-value deliveries* of software, as well as *loss of productivity* and *business loss*. However, this is rare due to a collaboration that the developers express as well functioning and satisfying for themselves, as well as for the customer.

5.3 Contingency

To deal with the causes and consequences of collaboration problems with the customer the developers at Bouvet have different short-term and long-term strategies. The list of contingencies is quite long, and the interviewed developers frequently use many of them in their collaboration with the customer. A lot of the contingencies found were not included in Hoda et al. (2011), including: *Building trust, self-testing, fill up backlog* and *allocate resources to other projects*. While some of the strategies included in Hoda et al. (2011) was not expressed as a practice at Bouvet. This included: *Changing customers' mind-set, providing options, Extreme Undercover* and *risk assessment up front*. Many of the strategies most used by Bouvet are linked to the *lack of time commitment* from the customer, as discussed earlier. These are especially: *Building trust, self-testing, fill up backlog* and *changing priority*.

In being involved with a busy customer, the development team at Bouvet have been forced to find alternative solutions to move forward in development. The problem is when strategies are used that break with core Agile principles. Some of Bouvet's strategies and work processes are in the grey zone of what can be called Agile development. In J. Highsmith and Cockburn (2001) it is stressed that there should be frequent and recurring check-ins and *just demos* with the customers. *Self-testing* of software is an example of a strategy that Bouvet is using that question their validity as an Agile team.

The strategy *self-testing* means that the team is not able to get the new functionality in front of the customer and ends up having to test the software themselves. Allowing the developers to test software for them is a sign of trust from the customers. On the other hand, extensive *self-testing* means that the team is falling back to principles that lie closer to Royce (1970)s Waterfall method than Agile principles. The whole idea of Agile methods is to consistently expose the customers and users of the system with working and valuable software. It is not the developer's fault that the customer is too busy

to give feedback, but they need to establish the routines with the customer that allow them to get continuous responses.

The first routine that Bouvet needs to look into is their ways of performing demos. Having state-of-the-art software that allow them to push finished software out on a staging environment, where the customer can test remotely, is a great tool. However, conducting a majority of customer testing remotely is not in-line with Agile principles. One of the core values of Agile is: “Individuals and interactions over processes and tool” (Beck et al., 2001). Bouvet should consider having more and recurring live demos with the customer. That is, as mentioned earlier, having a fixed Sprint period ending with a live test session with the customer. In this way, they can get the real-time impressions from the customer, as well as engage the customers on a completely different level than by having the customer testing remotely.

As mentioned earlier, *low-value deliveries* to the customers in Rovbase and Feltdagbok can be fatal. Getting live-demos also improves the chance that potentially hazardous bugs or errors are weeded out early. Even though Bouvet need to improve regarding their remote collaboration, this is also one of their strengths. Due to a lot of focus on *building trust* with their customer they communicate very well using *e-collaboration*. They also have a focus on getting the customer on the phone, rather than just sending emails back and fourth. As pointed out by the developers; discussing issues face-to-face or over the phone allows for real-time discussions, instead of emails that pile up in the inbox.

Another Agile value is: “Responding to change over following a plan” (Beck et al., 2001). According to J. Highsmith and Cockburn (2001) this means that the Agile team needs to be maneuverable. This is a value that Bouvet follows by being flexible with *changing priorities*. Even though Bouvet have many tasks lined up, they are not afraid to take a complete U-turn if necessary. There is definitely a need for the developers in this project to be maneuverable. Firstly, the customers have governmental rules and regulations that can sometimes change without much notice. They are also dealing with

multiple customers, which means their level of flexibility must increase as well. Again, the interviewees point out that *building trust* and establishing close communication is absolutely essential to allow rapid change. This includes trust between the Bouvet and their customers, but also between the customers themselves.

To enable flexibility, it is also an advantage to always have work to fall back on. Hence, to secure a constant workflow Bouvet make sure to *fill up the backlog* with tasks. By doing this Bouvet is showing understanding towards the customer's busy schedule. If the customer is right in the middle of a hectic period, the developers can ease off and work from the backlog. This a great way for the developers to avoid idle time and ensure productivity at all times. On the other hand, the developers have indicated that having too many open tasks can in fact be counterproductive. When new tasks are started, without old tasks being complete, there are just too many open loops. The developers can lose track of the progress and what the task actually entails. This means that re-work must be done, and thus productivity decreases.

Another way the developers can fall back in case the customer is busy is by *allocating resources to other projects*. If for example the developers in Rovbase need the customer for progress and he is unreachable, the customers can flex to other projects, while they wait for a reply. Unfortunately, this only increases the risk of the developers juggling too many open tasks. Also, the developers need time to familiarize themselves with the tasks in the other project, which again negatively influences productivity. Flexing to other projects gives some new opportunities though. In case one of the projects notices that they are running the risk of missing a deadline, they can allocate resources from other projects to strengthen the team towards the deadline. However, this should be viewed as a backup solution. It is more productive and comfortable to set *buffers* on intended deadlines. A strategy which is also commonly used by the interviewed developers.

There are many correlations with strategies found in this thesis and the ones in Hoda et al. (2011). However, this thesis has identified some new strategies,

and some of the strategies in Hoda et al. (2011) were not notably used by the developers at Bouvet. These are mostly linked to customers being sceptic about Agile software development. If the customer is sceptic developers can try; *changing the customers' mindset, providing options* or use Agile behind their back (*Extreme Undercover*). However, there was not much need for the use of these tactics from Bouvet. The developers indicated that the customer representatives were either very enthusiastic about working in Agile or didn't really notice what development method was being used. A customer that is typically resistant to change, as described by Hatling and Sørensen (1998), is in many ways a stereotype. It might still be common that customers are conservative, but there also exists customers that are hungry for change and improved development principles.

To sum up, Bouvet have a strong arsenal of strategies to deal with their customer collaboration. Bouvet implements the strategies through the trust and communication that they have established after a long-term collaboration with the customer. Hence, such strategies as digital communication and changing priorities on tasks, work well in the project. Unfortunately, the long-term trust between Bouvet and the customers can also lead to work habits that do not follow Agile principles. The lack of live demonstrations with the customer is one of the most noticeable effects. The developers fall back on *self-testing* the functions or having the customers perform remote tests on the system. To keep delivering high-level software, it is important for Bouvet to maintain best practices such as live demos. Even though they might be a little more time demanding than *e-collaboration*.

5.4 Summary

In this chapter the discussion was presented, comparing the empirical data with existing theory. Firstly, empirical results indicate that causes for developer and customer collaboration problems in the case project is due to a very busy customer. Also, the customers governmental organizations, making the customer situation quite complex. Secondly, the consequences of having a hard to reach customer is a lack of feedback. This leads to *low-value deliveries*

of software, potentially affecting the customer satisfaction level. Thirdly, empirical results point to the developers using an extensive set of strategies to deal a low responsive customer. This includes a considerable amount of *e-collaboration*, *filling up backlog* and *self-testing* features. Overall, the development team at Bouvet shows a well-developed customer relationship. Over time, mutual trust has been developed, enabling the developers to skip a check-in with the customer in decision-making. However, the lack of fixed Sprints and live demos are in direct conflict with the values of Agile software development. Without more face-to-face time with the customers the developers run the risk of creating software that is not in line with the customers wants and needs. Hence, Bouvet already have a strong and incorporated collaboration with their customer, just a few modifications to their Agile approach has the potential improve it even further.

6 Implications and Further Research

In the previous chapter, the findings were presented and discussed in light of the theoretical background. The following chapter will suggest implications and further research based on the discussion from the last chapter.

Implications will suggest ways this research potentially could impact theoretical, political and practical areas. The implication section will be followed up with a section on further research. This is a suggestion of how the content of this thesis can be studied and researched going forward, to qualify and further develop the core findings.

6.1 Implications

6.1.1 Theoretical

When looking at theoretical implications, it is relevant to compare the findings in this thesis with Hoda et al. (2011). First of all, most of the causes, consequences and contingencies presented in Hoda et al. (2011) have been indicated as relevant by the interviewed developers in this research. The amount of focus on customer relationships by the interviewed developers, together with the importance of collaboration in Agile values, underlines the need for theoretical contributions and advancements in this research area. Despite similarities, there are also several of the causes, consequences and contingencies found in Hoda et al. (2011), which were found to be less relevant in this thesis. This included *scepticism* from the customers, incidents of *business loss* and the need for *changing the customer's mind-set*. Hence, the customers were more open towards Agile than customers in previous literature. The thesis also introduced some new and unique causes, consequences and contingencies. This included, amongst other; *governmental issues*, *low-value deliveries* and *building trust*. Hoda et al. (2011) lacks to mention these incidents as weaknesses. This underlines the need for more research on this topic.

The contribution of this thesis is therefore to shine light on the range and severity of collaboration problems between developers and customers. The causes of collaboration problems are several, but the lack of time from the customer stands out. This is again the source of many consequences that can lead to slow and poor deliveries. The developers have a long list of strategies to work around these problems. The problem is that some of these strategies limit the quality of customer participation. This thesis shows that companies might be stating that they are using Agile principles but are in fact using a mix between Agile software development and the Waterfall method. I.e., the developers use a “*Scrum, but...*” methodology, not following through on the principles. The findings in this thesis suggest that theory needs to be even more critical on whether or not Agile teams are following Agile principles. Overall, the findings indicate that causes, consequences and contingencies behind developer and customer collaboration is a fairly untapped research topic.

6.1.2 Practical

Moving over to practical implications there are several ways software organizations, Agile teams and developers can learn from this thesis. Using the above theoretical implications, Agile practitioners can potentially improve both their collaboration and communication with the customers. Firstly, it is important that all stakeholders involved in the project are sufficiently informed and coached in their responsibilities in the project. Especially the customer needs to know their responsibility as an Agile customer. They need to know that Agile development can give them a product that covers their needs. But to do so they need to be involved throughout the process. Of course, the developers can choose to tailor certain parts of the process around the customer. However, certain processes are essential to achieve the full potential of Agile software development. One of the processes that is essential to be clear about, and properly teach the customer, is fixed Sprint lengths and live demos. It is important for the developers to signal the expectations of the customer representatives to the decision makers in the customer organization. Managers in the customer organization must set of time were customer

representatives can participate in the development process with input and feedback.

Even though following certain Agile principles is necessary, the developers must also practice flexibility and be open to changes. As the name “Agile” implies Agile software development embraces rapid change, rather than following processes or plans. These properties have made Spotify and other Agile companies able to compete with Goliaths’ such as Apple, Google and Amazon. Agile is built around the need for customers to respond quickly to market demands and changes. Hence, long-term and detailed development plans need to be minimal. Developers must instead focus on delivering working software to the customer iteratively so that the customers and preferably end-users receive high-value systems. The challenge is finding the right level of customer involvement, as well as the right balance of flexibility and structure. The most important takeaway for Agile practitioners is constant trial and error. Like Toyota and lean manufacturing, they must constantly experiment to enhance the process, and hence, keep striving for continuous improvements.

6.1.3 Political

Lastly, when looking at political implications, it is important to give some pointers to learning organizations. Regarding universities, existing literature and the research done in this thesis indicate that Agile methods should have a secure place in the curriculum of computer science. The students need to get comfortable with collaborating closely with customers. Firstly, having projects where computer science students cooperate with other study programs and professions is important. This is because customers will often communicate differently than software developers. Further on, in courses where students have the possibility to cooperate with real companies and software projects, it is important that they get an impression of how developers use Agile and the challenges involved in user/customer participation. Like programming, the tools of Agile methods are best learned by doing, rather than from a lecture or a textbook.

6.2 Further Research

Agile teams have been an increasingly trending research topic in the last decade. Research has also been done in the collaboration between developers and customers in Agile projects. As mentioned earlier though, the research done on developer and customer collaboration in Agile software projects is limited. However, papers like Kautz (2011) and Hoda et al. (2011) are examples of research extensively focusing on this topic. As with these papers, this thesis highlights the importance of further research on developer and customer collaboration. Agile leads to a higher demand for customer participation. Hence, increasing the use of Agile software development in projects will also increase the need for functioning collaboration between customers and developers. This again sets demand for further research on the causes and consequences of customer collaboration problems, as well as strategies to work around them. Hence, it is recommended that even more researchers address the need for more studies on this topic.

As mentioned earlier, in that Hoda et al. (2011) was a starting point for this thesis, the two share many similarities in identified causes, consequences and contingencies. However, the two also have many differences in research results. The differences were that several new causes, consequences and contingencies were found in this thesis. Also, this thesis introduced an indicator for the relevance of each cause, consequences and contingency. This showed that not all causes, consequences and contingencies were necessarily relevant to the Bouvet case project in this thesis. Given the number of differences between this thesis and Hoda et al. (2011) it is clear that further research on this topic must be applied to qualify and to elaborate on the identified differences. This is essential to enhance the understanding and further advance the research area of developer and customer collaboration.

To get a holistic and thorough research basis on customer and developer collaboration, it is essential for further research to test many different types of projects and participants. First of all it is recommended to include customers as active research participants. Studies on collaboration have often taken the viewpoint of either the developer or the customer. By including the customer

and developer in one study, it is possible to get opposing views, which might spur new findings. In the same way, by researching two different case companies, and comparing them together, there is a potential for new insights. Preferably the two companies are significantly different from each other, either in size, sector or geography.

In further research, it is also essential to choose different types of developers and customers. The developer in this thesis was a software consultant, and the customer was a governmental organization. This was a business to business (b2b) relationship, the organizations studied were Scandinavian and also well established. Firstly, studying a business to consumer organization, like Spotify, is an interesting research participant. This will likely affect the relationship the developers have with the customers, since they are developing directly for the mass market. Secondly, it is interesting to include a variety of companies, i.e., small and large as well as both entrepreneurial and established companies. Lastly, including developers and customers from other regions is recommended. Hoda et al. (2011) studied developers from New Zealand and India, and found many differences in their development and collaboration mentalities. There are many combinations of developer and customer types, again indicating the need for further studies in this research area.

6.3 Summing Up

Summing up, it is important to underline that teamwork and collaboration between two parties in computer science, and other professions, have never gone flawlessly. Stakeholders will never align one hundred percent, and agree on everything. The challenging part with Agile is that it increases the need for close and frequent participation from both the developer and customer. To get a functioning collaboration, the most important thing is trust. Trust is both indicated by the interviewed developers at Bouvet and previous literature, as a key property in any collaboration. A closer relationship only increases the need for trust. To have a successful Agile project, all stakeholders need to be

on the same page from day one. This means clear communication from both the developers and the customers on expectations from each other. Bouvet has built a great deal of trust with the customer. But they are not active enough in communicating and clarifying the development routines and deliveries. This falls negatively back on both parties. The customer gets dissatisfied because the software is not what they asked for, and the developers get a dissatisfied customer. This thesis identifies a need for researchers to keep evolving the research area and find ways to further improve the collaboration between developers and customers. In today's fast moving society, we must work iteratively and close to assure safe and reliable systems that are based on customer needs.

References

- Agency, N. E. (2015a). About Norwegian Environment Agency. from <http://www.miljodirektoratet.no/english/>
- Agency, N. E. (2015b). Om Rovbase. from <http://www.rovbase.no/Contentpages/InnsynOmRovbase30.aspx?Arbeidsomrade=InnsynOmRovbase30>
- Agency, N. E. (2015c). Rovbase. from <http://www.miljodirektoratet.no/no/Tjenester-og-verktoy/Database/rovbase/>
- Ahmad, M. O., Markkula, J., & Oivo, M. (2013). *Kanban in software development: A systematic literature review*. Paper presented at the Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on.
- Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). Manifesto for Agile Software Development. from <http://agilemanifesto.org>
- Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5), 30-39.
- Bouvet. (2015). About Bouvet. from <http://www.bouvet.no/en/About-Bouvet/>
- Brønnsundregistrene. (2015). Nøkkelopplysninger fra Enhetsregisteret. from <http://w2.brreg.no/enhet/sok/detalj.jsp?orgnr=974442167>
- Brustein, J. (2014). Spotify Hits 10 Million Paid Users. Now Can It Make Money? . from <http://www.businessweek.com/articles/2014-05-21/why-spotify-and-the-streaming-music-industry-cant-make-money>
- Cockburn, A., & Williams, L. (2003). Agile software development: It's about feedback and change. *Computer*, 36(6), 0039-0043.
- Cohen, D., Lindvall, M., & Costa, P. (2004). An introduction to agile methods. *Advances in Computers*, 62, 1-66.
- Dingsøy, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221.
- Dybå, T., & Dingsøy, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9), 833-859.
- Frederick P. Brooks, J. (1987). No Silver Bullet Essence and Accidents of Software Engineering. *Computer*, 20(4), 10-19.
- Glaser, B. G., & Strauss, A. L. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*: Aldine de Gruyter.
- Hatling, M., & Sørensen, K. (1998). Social constructions of user participation *The Spectre of Participation: Technology and Work in a Welfare State* (pp. 171-188): Scandinavian University Press.
- Highsmith, J. (2002). *Agile software development ecosystems*: Addison-Wesley Longman Publishing Co., Inc.
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120-122.

- Hoda, R., Noble, J., & Marshall, S. (2011). The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology*, 53(5), 521-534.
- Hope, K. L., & Amdahl, E. (2011). Configuring designers? Using one agile project management methodology to achieve user participation. *New Technology, Work and Employment*, 26(1), 54-67.
- Jeffrey, L. (2004). *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*: McGraw-Hill, New York.
- Kautz, K. (2011). Investigating the design process: participatory design in agile software development. *Information Technology & People*, 24(3), 217-235.
- Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum-making the most of both*: Lulu. com.
- Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, 36(6), 47-56.
- Linberg, K. R. (1999). Software developer perceptions about software project failure: a case study. *Journal of Systems and Software*, 49(2), 177-192.
- Martin, A., Biddle, R., & Noble, J. (2010). An ideal customer: a grounded theory of requirements elicitation, communication and acceptance on agile projects *Agile Software Development* (pp. 111-141): Springer.
- Martin, R. C. (2003). *Agile software development: principles, patterns, and practices*: Prentice Hall PTR.
- Merisalo-Rantanen, H., Tuunanen, T., & Rossi, M. (2005). Is extreme programming just old wine in new bottles: A comparison of two cases. *Journal of Database Management (JDM)*, 16(4), 41-61.
- Oates, B. J. (2005). *Researching information systems and computing*: Sage.
- Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3), 303-337.
- Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*: Addison-Wesley.
- Rasmusson, J. (2010). *The agile samurai: how agile masters deliver great software*: Pragmatic Bookshelf.
- Rico, D. F., Sayani, H. H., & Field, R. F. (2008). History of Computers, Electronic Commerce and Agile Methods. *Advances in Computers*, 73, 1-55.
- Ries, E. (2011). *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*: Random House LLC.
- Rowinski, D. (2014). In A Change From The Steve Jobs Era, Apple Is Listening. from <http://readwrite.com/2014/06/03/apple-ios-8-customer-feedback-android-features>
- Royce, W. W. (1970). *Managing the development of large software systems*. Paper presented at the Proceedings of IEEE WESCON.
- Sager, I. B., Peter; Reinhardt, Andy. (1998). Steve Jobs: 'There's Sanity Returning'. from <http://www.businessweek.com/1998/21/b3579165.htm>
- Schwaber, K., & Beedle, M. (2002). *Agile Software Development with Scrum*: Prentice Hall.
- Schwartz, E. (2004). IT myth 5: Most IT projects fail. from <http://www.infoworld.com/article/2667144/technology-business/it-myth-5--most-it-projects-fail.html>

- Subramanyam, R., Weisstein, F. L., & Krishnan, M. S. (2010). User participation in software development projects. *Communications of the ACM*, 53(3), 137-141.
- Surveillance, N. N. (2015a). Om Feltdagbok. from <https://feltdagbok.miljodirektoratet.no/OmFeltdagbok.aspx>
- Surveillance, N. N. (2015b). Om SNO. from <http://www.naturoppsyn.no/om-sno/>
- Sutherland, J. (2014a). *Scrum: The Art of Doing Twice the Work in Half the Time*: Random House LLC.
- Sutherland, J. (2014b). Spotify's Secret for Competing with Apple, Amazon, and Google. from <http://labs.openviewpartners.com/spotify-great-agile-example-scrum-done-right/>
- Tjora, A. (2012). *Kvalitative forskningsmetoder i praksis*: Gyldendal Akademisk.
- Torstensen, A. (2014). User and Customer Participation in Agile Software Development. *Unpublished*, 40.
- Walsham, G. (1995). Interpretive case studies in IS research: nature and method. *European Journal of information systems*, 4(2), 74-81.
- Widding, L. Ø. (2005). Hovedutfordringer knyttet til ulike forskningsdesing når hensikten er å generalisere. *Unpublished*, 1-17.
- Widding, L. Ø. (2006). Teorigenerering basert på case-intervjuer. *Unpublished*, 1-14.

Appendix

A1 Interview Guide

The interview guide is presented in Norwegian, its original language.

Introduksjon

Forklar temaet: Meningen med oppgaven er å finne ut hvor utfordringen i samarbeidet mellom utviklere og kunder ligger, hva konsekvenser dette har og hvilket strategier som blir brukt for å håndtere utfordringene.

- Forklar at "Agile" er et paraply begrep for Scrum, XP, Lean, etc.
- Spør om noe er uklart og om respondenten har noen spørsmål.
- Informer om opptak og sørg for samtykke
- Om ønskelig kan transkribert innhold bli lest av intervjuobjekt

*Start opptak

Generelt

- Hva heter du?
- Jobber du med Miljødirektoratet som kunde?
- Hvilket Agile utviklingsmetode bruker dere?
- Hvilket Sprint perioder opererer dere med? (Kommentar: Hvor lang tid går det mellom hver Sprint?)
- Hvem er kunden og hvem er brukeren?
- Hvordan måler dere kundetilfredsstillelse og/eller brukertilfredsstillelse?
- Hvor mye kontakt har du med kunden?
- Opplever du utfordringer i ditt samarbeid med kunden? (ja/nei)

Grunner – I denne fasen skal vi fokusere på hvorfor det skjer utfordringer i samarbeidet med kunden.

- Har du noen tanker om hvorfor dere opplever utfordringer i samarbeid med kunden?
- Opplever du at kunden er skeptisk til Agile?
- Opplever du at distanse med kunden er et problem?
- Opplever du at kunden ikke har nok tid å sette av på prosjektet?

- Dette er et ganske stort prosjekt, opplever du at størrelsen på kunden avgjør hvordan de mottar Agile?
- Opplever du at Agile er utfordrende i kontraktforhandlinger med kunden? (Kommentar: Ift., at det er vanskelig å estimere et nøyaktig budsjett til kunden)
- Opplever du at ineffektive kunderepresentanter gjør Agile til en utfordring? (Eksempler 1: Gir ikke nødvendige krav og kommentarer. Eksempel 2: Forstår ikke godt nok hva Agile er)
- Har du noen andre kommentarer ift., grunner for utfordringer med kunden?

Konsekvenser – I denne fasen skal vi fokusere på konsekvenser av utfordringer i samarbeid med kunden.

- Har du noen tanker om hva utfordringer med kunden kan føre til?
- Kan teamet bli presset til å overprestere? (Eksempel: Gjennom en satt kontrakt eller et satt budsjett?)
- Kan det være problemer med å samle inn og avklare hvilket krav kunden har til systemet?
- Kan det være problemer knyttet til å prioritere krav?
- Kan det være problemer knyttet til å få nok tilbakemeldinger fra kunden på arbeidet?
- Kan problemer med kundesamarbeidet føre til tapt produktivitet?
- Kan problemer med kunden i ytterst konsekvens før til tap av kunder eller annen tap av forretning?
- Har du andre kommentarer ift., konsekvenser som utfordringer med kunden kan føre til?

Strategier – I denne fasen skal vi fokusere på strategier dere bruker for å jobbe rundt utfordringene med kunden.

- Har du tanker rundt metodene dere bruker for å jobbe rundt utfordringer med kunden?
- Prøver dere å endre kundens meninger om Agile? (Eksempel: Kunden ønsker en vannfallutvikling, men dere forklarer fordelene med Agile)

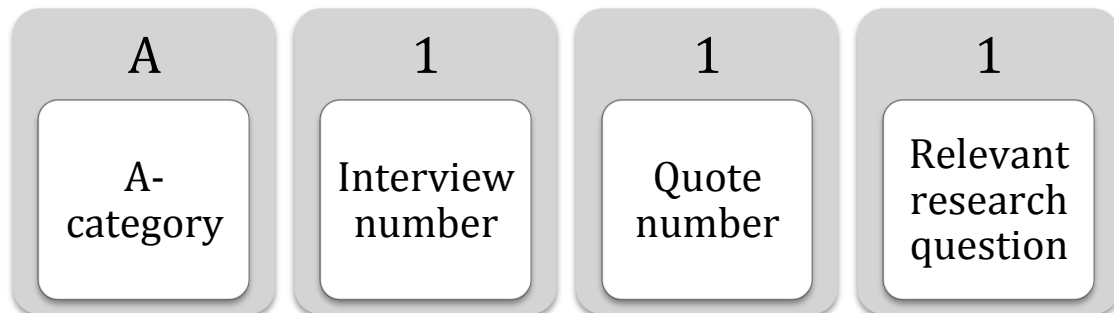
- Prøver dere å gi kunden alternativer, dvs., forme arbeidsmetodene etter deres ønsker?
- Bruker dere demoer av produktet for å engasjere kunden og på den måten sikre nødvendige tilbakemeldinger?
- Hender det at dere endrer prioritet på brukerhistorier for at det skal passe inn bedre med kunden?
- Gjør dere en slags risikovurdering med kunden før prosjektet? (Eksempel: Ber kunden fylle ut en spørreundersøkelse for å kartlegge deres behov?)
- Bruker dere noen gang Agile prinsipper uten å opplyse kunden? (Eksempel: Det har vært Agile bedrifter som gir kunden uttrykk for at vannfall blir brukt, men egentlig har de fylt Agile prinsipper)
- Opererer dere med en buffer på den estimerte tiden på prosjektet for å gi noe slyngningsrom? (Kommentar: Dersom det opereres med et fast budsjett eller en fast kontrakt kan det for eksempel lages et buffer i tilfelle "worst-case")
- Legger dere opp til en fordeling av kundene over forskjellige faser av prosjektet, istedenfor at det er en satt kunderepresentant for hele prosjektet? (Kommentar: En kunde er med en Sprint og etter den er ferdig kommer en annen kunde som er ansvarlig for neste Sprint)
- Bruker dere noen på utviklingsteamet som en kundeveileder, en som har hovedansvar for å koordinere kontakt med kunden? (Kommentar: Kalles gjerne "customer proxy")
- Bruker dere e-kommunikasjon (email, telefon, chat eller videokonferanse) for å kommunisere oftere og enklere med kunden?
- Har du flere kommentarer til strategier som dere bruker for å jobbe rundt utfordringer med kunden?

Oppsummering

- Er det noe du vil legge til?

A2 A-Categories

The A-categories represent direct quotes from the interviews with the developer team at Bouvet. The parts presented here were considered to have most relevance to the scope of the thesis. The codes can be read like this:



The interview guide is presented in Norwegian, its original language.

1 Kenneth, 26.02.2015

A-1-1-0

Jeg vil at vi bruker en blanding av Kanban og Scrum. Altså vi har mye av planleggingssyklusen finner man i Scrum-metodikken. Men, mye av den daglige driften, med oppgavefordeling og køfordeling og sånt, den er fra Kanban-teorien.

A-1-2-0

Vi prodsetter minimum en gang i måneden

A-1-3-0

Så er det da testmiljøet som er det offisielle som dem (kunden) kan gå inn å se på, så det de kan forvente seg å få av funksjonalitet. Det oppdaterer vi sikkert annen hver, tredje hver, dag.

A-1-4-0

Kunden er, Det Norske Miljødirektoratet i Norge, og det er Naturgårdsverket i Sverige.

A-1-5-0

Miljødirektoratet er en veldig veldig aktiv bruker av systemet

A-1-6-0

Det er ikke Miljødirektoratet som samler inn data, det er det fylkesmennene.

A-1-7-0

I Sverige er det lensmannen, altså fylkesmannen er jo lensstyrelse i Sverige.

A-1-8-0

Miljødirektoratet og Naturgårdsverket som konsumerer dataen, det er de som fatter beslutningene på grunnlag av alle dataene som kommer inn i systemet. Det er ikke det som står for datafangsten. Det er det de operative avdelingene som gjør, ute i naturen.

A-1-9-0

tross for at det er nå snart 2000 brukere av den fagapplikasjonen, så er det veldig mange av de brukerne som vet hvem jeg er, og vet mitt telefonnummer. Det er ingen utenom Miljødirektoratet og Naturgårdsverket som har lovt med å komme med bestillinger, men de har veldig lov med å komme med tilbakemeldinger og ønsker. Ønsker skal egentlig da rutes tjenestevei altså enten til Miljødirektoratet eller til Naturgårdsverket.

A-1-11-1

Vi jobber med produkteiere som er veldig travelt opptatt, som virkelig brenner for saken, og som jobber mye mer en normale arbeidstakere. Så det, litt på tilgang, tilgangen til produkteiere

A-1-12-1

litt på de utfordringene som naturligvis kommer når vi jobber med to forskjellige land. Vi får jo en del spørsmål om EU, ikke EU medlemskap, altså

Sverige er jo underlagt masse bestemmelser på grunn av de er medlem av EU. Så vi ikke har i Norge.

A-1-13-1

Nei, overhodet ikke, vi startet, så forløpet til Rovbase 3.0 het Rovbase 2.5. Det startet vi vell med i 2001, eller noe sånt. Og uten å vite det, så drev vi da med Agile utvikling. Så da jeg ble gjort oppmerksom på Agile utvikling, at det liksom var en allment akseptert standard, for å si det sånn, på en seminar i Oslo

A-1-14-1

Uten å vite det, så har vi både, altså, uten at hverken jeg eller Miljødirektoratet, det var jo vi som var de første partnerene. Uten at vi visste om teorien, så har vi drevet med Agile utvikling

A-1-15-1

A: Så du svarer tvertimot da?

K: Det er stor entusiasme for det.

A-1-16-1

A: Opplever du at distanse med kunden kan være en grunn til samarbeidsproblemer?

K: Nei, overhodet ikke.

A-1-17-1

Vi ringes til alle døgnets tider for å si det slikt. Vi har et veldig tett samarbeid. Vi kan jo nesten se at jeg og den norske produkteieren og den svenske produkteieren er veldig gode venner for å si det slikt. Vi kan godt spore av under kveldsmøtene og snakke om noe helt annet enn det vi skal snakke om. Nei, vi har, et veldig, vi har opparbeidet oss et gjensidig tillitsforhold.

A-1-18-1

Nei, det er så gode muligheter for å ringe folk på Lync eller Skype eller, vi har Tandberg-video-system her. Det er de også hos Miljødirektoratet, nei hos

Naturgårdsverket i Stockholm. Og, nei vi er så samkjørte og tenker så likt at det opplever vi ikke, distanse som noe problem.

A-1-19-1

Av og til så de så opptatt med andre ting, altså, de har en voldsom arbeidsbelastning, at det kan, det kan drøye med spørsmål som vi venter på avklaring om

A-1-20-3

Men, på den andre siden, så har vi da, vi har en god backlog. Så, det er, det skjer aldri at vi sitter å venter, altså at vi ikke har noe arbeidsoppgaver. Da setter vi bare den oppgaven på vent, så begynner vi på noe annet. Så det krever bare litt planlegging fra min side

A-1-21-3

Kunden er og veldig opptatt av kvaliteten i systemet, altså i koden, så vi har frie tøyler til å drive med refaktorering.

A-1-22-0

vi ligger på et samlet uttak på over, ja, på cirka 5 millioner i året. Så, sånn sett så er det, vi får gjort veldig mye for de 5 millionene for å si det sånn. Så, det er, så både, det er Miljødirektoratets mest virksomhetskritiske system. Det utbetales jo over en halv milliard kroner i erstatning bl.a., til sau- og reineiere.

A-1-23-1

A: Men det er jo en stor kunde. Opplever du da at størrelsen på kunden kan avgjør hvordan Agile fungerer i praksis? Eller hvordan dere får Agile til å fungere?

K: Nei, jeg synes, altså det går veldig bra, det er jo, i å med at kunden er delt opp i flere fragmenter liksom, så er dem veldig, alle som er med i diskusjonene og sånne ting er veldig flink til å skjønne sin rolle. Så, og skjønner det at vi driver med Agile, og dem har tilgang til Gira, dem legger inn, legger inn ønsker og bestillinger og de får feltrapporter som kommer inn og slikt. Vi tenker veldig sjeldent på om det er en sak som har kommet inn fra

Viltskadesenteret eller SMO eller Miljødirektoratet. Det er en felles tankegang om hva løsningen skal være.

A-1-24-1

Rovbase prosjektet har alltid vært drevet etter avtaler som støtter smidig utviklingsarbeid. Dagens forvaltningsavtale står det vell klart og tydelig at vi, for en gitt sum, skal drifte de forskjellige funksjonene som finnes i løsningen. Ikke noe om leveringstidspunkt, og spesifikk dato og kodelinjer som skal skrives eller noe sånt. Men, det baserer seg på tillit. Tillit, tillit og tillit for å si det sånn, de tre t-ene.

A-1-25-1

Vi velger å ikke se på det som en utfordring. Viss det er folk som for eksempel har ansvar for et av arbeidsområdene i løsningen, lærer seg for eksempel Fallvilt eller Fjellrev eller Rovviltobservasjoner, viss dem ikke leverer, så starter vi i samråd med produkteiererne bare på andre oppgaver. Så får de bare vente.

A-1-26-3

Da vil brukerne automatisk gå til de som er ansvarlige for de ulike arbeidsområdene og begynne å etterlyse; hvorfor skjer det ingenting? Så er det de som får ta det. Vi får levert uansett for å si det sånn. Og, så det er stor selvjusit, for å si det sånn. Viss dem ikke kommer med de nødvendige avklaringene, så får de ikke levert funksjonaliteten. Da går vi bare videre, så, og det lærer dem veldig fort. At skal dem ha levert sine ting, så må de selv levere.

A-1-27-1

altså, norske og svenske regler kan jo være litt forskjellige. Norsk og svensk utbetalingsregelment er veldig forskjellig. Der har vi valgt å holde det helt adskilt. men når det gjelder hvordan man overvåke de store rovdyrene, så har det og vært forskjellig metodebeskrivelser.

A-1-28-3

Det har vi faktisk nå klart å jobbet frem til at det er felles metodikk i Norge og Sverige. Og da får arbeidet mellom løsningene en ekstra dimensjon, når vi liksom klare å ikke bare levere en IT løsning, men vi klarer faktisk å få arbeidsmetodikken nå til å bli felles. Da bruker dem felles system i de to landene og så jobber dem på samme måte ute i felt. På grunnlag av at dataen skal inn i det system. Så da blir det dobbeltpositivetet, for å si det sånn.

A-1-29-2

det kan være satte datoer, for eksempel saueerstatningen og reinerstatningen skal behandles hos fylkesmennene til en gitt dato for da skal saueeierene og reinerne få brev om hvor mye penger de får. Så da er det, da er det tidspress

A-1-30-3

jeg tror ikke at noen føler at de må overprestere for det vi vet det, nå har vi drevet i det her gamet her i samme mange år. Så vi vet helt klart når de periodene, hvor det er veldig mye å gjøre kommer, slik at da er det min oppgave å rydde plass på tavlen, slik at det ikke står andre oppgaver på de personene som skal jobbe med det der.

A-1-31-2

For det kommer jo stadig, altså, hvert år kommer det jo nye erstatningsregler, nye kalkyler, og de kalkylene må jo implementeres og testes og testes hos kunde og, en lang løype der, det skal ny ordlyd ut i alle vedtekstbrev og slike ting. Så da, da setter vi av egentlig lange perioder hvor vi kun fokusere på det.

A-1-32-2

A: Forstår. Kan det være problem knyttet til, nå har vi jo vært litt borti det da, men å samle inn og avklare hvilket krav kunden har til systemet?

K: Det kan være det.

A-1-33-3

Bouvet og Naturgårdsverket og Miljødirektoratet har en antagelse om at det kan være vanskelig å få inn avklaringer, så samler vi rett og slett interessentene til en workshop, på 1 eller 2 dager. Hvor vi går gjennom og finner behov og skikker på mulige løsninger og slike ting. Og får spikra mest mulig, for kunden, produkteierene er jo veldig veldig opptatt på at ting skal ikke drøye ut i tid.

A-1-34-2

Nei, det går veldig bra, altså prioriteringene, det må jo veldig ofte prioriteres mellom norske behov og svenske behov og de felles behovene vi har. Så da, da snakker de forskjellige landene veldig godt sammen om hva som er viktig.

A-1-35-0

Rovbase har blitt tatt i bruk av svensk og dansk og finsk fagmiljø på grunn av at det er tradisjon for samarbeid.

A-1-36-2

Nei, jeg synes ikke det. Det er en dedikert gjeng. Som og er da lidenskapelig opptatt av løsningen, så det, det er en veldig stor interesse for å sjekke opp ting, og komme med tilbakemeldinger, så det, så er det jo det med selvjustisen at kommer dem ikke med avklaringer så blir de ikke med på neste release. Men, da er det dem som får spørsmålet.

A-1-37-2

Nei, ref. det at vi har en god og lang backlog, og stadig ny teknologi som skal implementeres, så har vi, vi har aldri et produksjons-, produktivitets-issue for å si det sånn.

A-1-38-2

Nei, det går veldig veldig bra.

A-1-39-2

dersom samarbeidsforholdet ikke fungerer så får man jo ikke ut nødvendig funksjonalitet, som da kan være målt oppimot tidsfrister da, som stortinget og regjering har fastsatt. Og da kan jo i ytterste konsekvens fylkesmannen bli saksøkt, for ikke å ha overholdt tilbakemeldingsdatoer og sånne ting. Og det er klart, er det feilaktig funksjonalitet i arbeidsområdet for duer, rovdyr, rovviltsobservasjons-individ så kan det jo bli fattet vedtak om utskyting av bjørn, som er rett og slett feil. Istedenfor at løsningen kunne vist at det var 20 bjørn i området, men de reelle tallet er 3. Og da skulle det ikke tas ut. Så det er store verdier på spill, både økonomiske og natur-, altså miljømessige verdier. Så, konsekvensen av et dårlig samarbeid kunne ha vært katastrofalt. Dårlig testing av kalkulasjonsmetodene for eksempel kan det gjøres feil utbetalinger

A-1-40-3

det jeg sier til alle som starter på teamet, det er det at, viktige, altså, avklaringer som haster skal ikke tas på email, de skal tas på telefon. Eller, nettmøtet. Så, gjerne oppsummer ting i en email, eller i Gira, altså koble opp den emailen i Gira for å ha dokumentert hva man ble enige om. Men, ring. Ring først. Da blir man kjent med dem som, så det kan være nye kunderepresentanter inn i systemet og. Og da blir man kjent, også får man avklaringen med en gang.

A-1-41-3

Petter Jaxgård i Sverige, han har lært seg user-stories og full pakke for, det er bare fryd og gammen for å si det sånn

A-1-42-3

Vi har vell egentlig funnet en samarbeidsform som, som fungerer. Vi prøver jo litt nye verktøy og slikt av og til, prototypingsverktøy i tidligere designfaser og slike ting. Men selve, men selve Agile metodikken går vell bare av seg selv. Så visst det skjer endring der så er det noe som vi i felleskap har funnet ut. Vi tok jo inn Gira, så vi brukte noe Microsoft verktøy før, synes ikke at det fungerte på slutten da, og da tok vi inn Gira for noen år siden. Og da var det bare liksom, vi bare fant et verktøy og sjekket det ut med produkteierne, om de

synes det var bra, og at de fikk skrevet inn nye "tasker" og kunne følge saksgangen og slikt.

A-1-43-3

K: Ja, det må jo tilpasses. Av og til er det langsiktige arbeidsområder vi jobber med, og av og til er det veldig veldig hastende.

A: Ja, men det tilpasses gjerne etter kunden.

K: Etter kunden, ja.

A-1-44-3

K: Alle dem som har bestemmelsesmyndighet i prosjektet skal kjenne til hele løsningen. Da må vi vise frem, altså vi har dem alltid med i møter og viser. Det er jo for å lære dem opp i systemets kompleksitet også.

A: Ja, gjerne for å engasjere de også, til å dele.

K: Jaja, og for å få nye idéer til sine arbeidsområder. I det datasettet blir det gjort på den måten, og det kan jo fungere, ja, sånn vil vi ha.

A-1-45-3

A: Og hender det at dere endrer prioritet på brukerhistoriene basert på tidsplanen til kunden?

K: Det skjer veldig ofte.

A-1-46-1

Kunden kan jo fort presentere nye retningslinjer fra stat og regjering. Og da er det, det må bare overholdes.

A-1-47-2

Det er jo veldig ofte at det kommer, det kan komme en lovendring i mars, så lovendringen blir ferdig vedtatt i mars og den er gjeldende fra første januar. Da, da er vi liksom to måneder på etterskudd med en gang, og da må det prioriteres.

A-1-48-3

Og det har, visst det kommer en slik endring i Norge så har Sverige forståelse for at selvfølgelig må ressursene settes der. Og visa versa.

A-1-50-3

Nei, vi viste oss tilliten verdig for å si det sånn.

A-1-51-3

Nei, vi er alltid åpen om at vi bruker Agile.

A-1-52-3

Vi legger på alle faktorer, kjent sykefravær og ferieavvikling og den type, den type ting. Så vi har jo store erfaringstall så vi klarer å være veldig godt innenfor der, for å si det sånn.

A-1-53-3

Ja, veldig. Nå driver vi med, ja, konkret nå da, så driver vi med fjellrevfunksjonalitet for tiden. Og det er da en seksjon i Miljødirektoratet og Naturgårdverket som liksom styrer med. Så har de da knyttet til seg annen interesser, og som er med i det. Når det blir ferdig så skal vi inn med funksjonalitet for jerv, og det er da annen faggrupper som er inne igjen, så vi, det er veldig mye folk, til og fra, for å si det sånn.

A-1-54-3

K: Ja, men det er vell egentlig, det er vell jeg som tar den rollen. Ihvertfall på det som går på domenekunnskap. Brukskvalitetsgjengen vår er veldig flink til å ta den rollen når det, når det er nytt brukergrensesnitt som skal avklares. For da, da er jeg gjerne koblet lite inn i den prosessen som. Så får de, så får de styret det. Altså, jeg er med å koordinere, men de kjører det løpet selv. Men det er vell som er, altså, det er vell jeg som er kundekontakten.

A: Og, du har vært inne på det litt før. Men, bruker dere e-kommunikasjon som email, telefon, chat, videokonferanse for å kommunisere oftere og enklere med kunden.

K: Ja, hele tiden.

A-1-55-3

Ikke annet enn at åpenhet, ærlighet, personlig kontakt istedenfor en email som gjerne kan ligge å vente litt.

2 John Sverre, 26.02.2015

A-2-1-0

Ja, det er litt sånn i grenseland tror jeg egentlig, men det er vell en plass mellom Scrum og Kanban. Kanskje mest litt Kanban, kan man si. Vi reindyrker ikke noe spesielt, men nå var det ganske tilpasset da, så, er vell at det passer veldig bra for vår arbeidssituasjon.

A-2-2-0

Ja, nå vet ikke jeg om vi kan kalle det for Sprinter. Vi har heller, forsåvidt, det er jo prioriteringer av oppgaver fra backloggen som plukkes ut, kan du si. Og så, er det, veldig individuell vurdering, kan du si, alt etter prioriteten på oppgavene som ligger i det utvalget som avgjør hvor lang perioder vi utvikler før vi faktisk pusher det til kunden da.

A-2-3-0

Så, vi får prioriterte oppgaver i samarbeid med kunden, som sier at dette her må på plass, og kanskje gjerne har en tidsfrist også da. Og da blir jo de oppgavene på en måte pushet gjennom løypen, så fort vi får det til egentlig. Men, ellers så, vi har da lengder på en måned kanskje, som normalt da, mellom hver produksjonssentring.

A-2-4-0

Vi har visninger ja, men ikke noe sånn møte opp hos kunde, og holder noen demoer, offisielt, eller noe sånt nei.

A-2-5-0

Men så har vi på en måte, det som staging serveren da, som tester ut, vi kaller det bare test server. Der vi legger ut alt som et forsteg til produksjon. Hvor

kunden går inn og verifisere, eventuelt at dem godkjenner eller sender det tilbake, på en måte da.

A-2-6-0

Kunden for vår del er jo Miljødirektoratet, og spesielt på Rovbase så er det Viltseksjonen, som står som kunde der. Brukeren, der er jo mange flere. Det er jo blant annet de ansatte på Miljødirektoratet, men også alle i SNO som, på en måte, oppsyn som er underlagt av direktoratet. Og så har vi jo nå i senere tid, de par siste årene, så har Sverige vært med også, Naturgårdsverket, som har vært både kunder og brukere. På samme som Norge er med da. Med sine egne oppsyn da, så du kan si, da ble det gjort et spleiselag da, mellom Naturgårdsverket og Miljødirektoratet som, på en måte, oppdragsgivere her. Så kommer begge med sine brukergrupper da.

A-2-7-0

Det er, kan du si, det er jo mange måter, direkte tilbakemeldinger er kanskje den viktigste da. Men, det er jo gjennom at vi har 2 kunder, kan du si, så blir det jo litt sånn splittet opp. Det bruker jo å holdes sånn planleggingsmøter, der det evalueres litt frem og tilbake.

A-2-8-1

Ja, du kan jo si at, det er kunder som har veldig mye å gjøre da. I den saksbehandlingen som de driver med til daglig. Slik at, kanskje ofte, så blir den videre utviklingen av det systemet sekundær i forhold til det arbeidet de holder på med, eller jobber med da. At dem ikke er 100% ansatt for utviklingen av dette her systemet. Det blir mer et støtteverk for den jobben dem gjør, så da blir det en litt skjevfordeling da. Så det vil jo si at kunden ikke er tilgjengelig alltid, når vi gjerne skulle hatt kontakt da. Så der, der er jo kanskje den mest klare utfordringen vil jeg si.

A-2-9-1

Jeg tror det er foretrukket fra deres side, så den arbeidsmodellen som vi har kommet frem til i Rovbase ihvertfall, den tror jeg framheves som ganske

suksessfull i denne sammenhengen. Så, de er stolt over hvordan det drives, og veldig fornøyd med det arbeidet som gjøres der da.

A-2-10-1

Nei, det er jo ikke noe distanse å snakke om da.

A-2-11-1

A: Nå som Sverige er med, er det noen utfordringer i forhold til distansen der?

J: Nei, de er veldig flinke til å kommunisere, så ofte, så de er mer tilgjengelig de, enn den norske kunden da. For de, til gjengjeld er de desto mer ivrig på det digitale media, kan man si da. Slik at kommunikasjonen med dem er enkel å opprettholde likevel da.

A-2-12-1

Jeg vet ikke om jeg kan trekke noen link der egentlig. Det går jo litt mer på, kanskje på de som produktet er komfortabel med egentlig da. Så, de det ikke angår på en måte, de er jo utenfor loopen uansett. Så selv om kunden er stor, så kanskje ikke de involverte partene er like mange.

A-2-13-1

Men i og med at Rovbase er et forvaltningsprosjekt, så er det jo ikke de samme begrensningene der heller egentlig. Den Innkjøps prosessen kan man si, der man vil ha en pris på et produkt, og den faller litt bort da. Da er det snakk om at vi vinner en kontrakt på en tre-årsperiode som inneholder vedlikehold av systemet. Og da har jo den et pris ifra deres side.

A-2-14-1

istedenfor at, hvor mye kommer dette produktet til å koste, så er det mer hva får vi gjort innenfor dette beløpet her, men det vet jo man uansett ikke, kan man si. For der er det jo den smidige parten som kommer inn der, som var veldig grei da. I at, oppgavene er jo ikke definert på forhånd, de defineres jo underveis, og så løser vi dem bare.

A-2-15-1

Nei, jeg tenker kanskje at Agile fungerer bedre i den situasjonen da. Fordi at, da behøver man ikke nødvendigvis å spesifisere så godt som kanskje man måtte gjort ellers da. Så, selvfølgelig, det finnes områder der kunden eller kanskje enkelte ressurser hos kunden ikke har like stor kompetanse på. Men, da tar man jo bare ekstra runder og kommer, kommer frem til forståelse om hva man skal gjøre da. Så, det synes jeg ikke er noe problem nei.

A-2-16-3

Ja, altså, i og med at vi har en så stor backlogg som vi har, så kan du si, vi får alltid til å gjøre noe, og vi får alltid til å levere noe.

A-2-17-2

Men, det er kanskje ikke det som kunden trenger mest da. Så det vil jo være en konsekvens, at dersom kunden i perioder ikke involvere seg aktivt, så får de levert funksjonalitet som kanskje ikke var det dem trengte mest på det tidspunktet.

A-2-18-2

Men også når dem da har prioritert at dem er med og spesialisere dem oppgavene som var valgt ut da. For vi er jo avhengig av at de spesifisere, både fordi at det er ganske, kan man si, komplekst arbeidsområdet. Der vi ikke nødvendigvis har kunnskap, men også fordi det er jo ikke vi som skal ønsket for dem, hva de vil ha da. Men at det skal jo tilpasses deres arbeidsflyt, så godt som mulig. Så dersom det leddet man mangler da, at dem ikke er med aktivt og sier at, sann vil vi ha det, så kan det være at vi må skyve på den oppgaven da, og ta opp andre oppgaver som kanskje vi vet hvordan skal gjøres istedenfor da. Så, men i og med at, som sagt, den backloggen er så kjukk, og vi har to kunder som begge er foran med behov, så, så vil vi alltid ha noe å gjøre da.

A-2-19-2

Deadliner, det er ikke så mye deadliner for vår del

A-2-20-1

La oss si at det kommer en forskrift ifra høyere hold, som må på en måte innfris inn i produktene her, fordi det er jo statlig. Og det inngår jo på slike ting som erstatning, og erstatningssatser, og slike ting. Så da kan det jo være at, ja innenfor, innenfor 1. november så skal den nye søknaden kunne være med, så sånne ting som det der da.

A-2-21-2

Så, vi vil jo alltid ja frister og sånt da. Og, ja, det er vell det at, det har jo selvfølgelig en tendens til å bli mer hektisk inn mot slike frister da. Men, jeg vil ikke si at det er snakk om noe overprestering på noen måte.

A-2-22-2

Så lenge mann har den kommunikasjonen mann har med kunden og i tid, i god tid foran der igjen da, så klarer du å møte de fristene. Men selvfølgelig, viss det her er perioder som kunden samtidig er utilgjengelig, av de samme grunnene som vi snakket om tidligere da, så vil det jo være vanskelig for oss å møte den fristen. Men som er det ikke vi som skal overholde de fristene heller, så viss vi gjør vårt beste på en måte da, men det er kunden som ikke kan møte tidsfristen på grunn av deres eget fravær, da er det jo for det første er det ikke ansvaret våres, og for det andre så er det jo forsinkelser da som ikke hadde noe med det arbeidet vi gjorde, men den responsen vi fikk fra dem kanskje da.

A-2-23-2

Vi får ikke skylden uansett, men selvfølgelig, vi jobber jo for at vi skal møte de fristene som er satt da. For det vil jo selvfølgelig slå tilbake på oss uansett, dersom produktet vi leverer ikke innfrir.

A-2-24-2

Ja, det er jo en litt sånn spesiell situasjon, i og med at vi har to kunder her da. Fordi de har jo ulike, ulike behov til å begynne med. Det var jo opprinnelig et norsk system, som var tilpasset norske arbeidsmetoder, og så kommer jo svenskene inn som hadde et system fra før, som fungerte på en egen måte. Og så skal de over på det systemet vi har, der det nødvendigvis må en del

justeringer til da, og kompromisser. Så, sånn, selvfølgelig, da kan det jo bli fort interessekonflikter da, i forhold til hva er det mann faktisk vil ha. Vil mann tilpasse det mer likt den svenske modellen, eller vil man beholde den norske modellen, slik som den var. Så, der kan det jo være at kravene vil ha ulike, kan mann si, interessentene har sant, ulike perspektiver som gjør at kravene vil se forskjellig ut.

A-2-25-2

Kanskje også prioriteringene er forskjellig, så der er det jo viktig da, at de blir enige seg imellom om hva som er viktigst og ikke.

A-2-26-2

Ja, det blir jo samme grunnen da, noe som brenner i Norge, der ikke sikkert det er så viktig for svenskene. Så, det blir jo et slikt vektet produkt, den prioriteringen der. Utifra at vi har jo to kunder, der vi skal prøve å holde begge fornøyd. Så vårt ansvar overfor dem er jo at vi må prioritere litt fra begge. Men så er det jo et samarbeid imellom dem da, så, i tillegg til at vi prioritere for dem, så må også dem sammen se på det. Men selvfølgelig det er jo utfordringer knyttet til det.

A-2-27-2

Ja, det er, det er jo, ulik grad kan mann si da. Viss at det er perioder hvor kunden er veldig travel så vil det jo selvfølgelig også være tilbakemeldinger, og ikke bare kravspesifisering som er fraværende. Det er bare å tenke seg at situasjonen der vi setter ting i produksjon som vi ikke er så, så vell testet som det skulle ha vært, det vil jo være et resultat av at kunden ikke er så tilgjengelig som han skal. Vi har en frist vi skal møte og viss vi ikke da får nok tilbakemeldinger, så må vi jo bare gå utifra at det er godt nok.

A-2-28-2

Ja, dersom kunden ikke er aktiv, så fører det jo til at vi sitter litt på gjerdet. For da må jo vi vente unødig på tilbakemeldinger, og det blir mye mer kommunikasjon som ikke hadde behøvd å finnes. Som purringer på tilbakemelding, og kanskje vi må teste mer selv, og hopp bukk over ledd og

prøve å kontakte andre personer som kan gjøre jobben for oss. Så, det blir jo en del "overhead" der ja, og det er ikke sikkert at vi alltid har oppgaver som passer, kan du si, inni. Det er alltid nye oppgaver å gjøre, men det er ikke sikkert at den passer å begynne på akkurat der og da. Slik at, det blir rett og slett, dødtid. Så det fører jo til lavere produktivitet hos, for kunden da, kan du si.

A-2-29-3

på grunn av at vi flekser mellom andre prosjekter her, slik at vi klarer alltid å få flyttet om dersom det blir litt dødtid på et prosjekt. Vi har jo vært inne på flere prosjekter innenfor det teamet vi har, slik at da flytter vi våre ressurser imellom de prosjektene. Så det er ikke noe problem for oss, men det blir jo selvfølgelig opplevd lavere produktivitet ovenfor dette her enkeltprosjektet for kunden da.

A-2-30-2

For Bouvet så vil det jo kunne føre til at vi ikke når opp til taket i kontrakten, som gjør at, for vi fakturere jo løpende. Det er jo ikke, vi får ikke beløpene som kontrakten er satt på. Slik at, dersom det blir mindre arbeid i kontrakten vi gikk på, så mister vi jo de. Så du kan si, der er det jo selvfølgelig, det kan føre til tap.

A-2-31-0

Så, men akkurat i dette tilfelle så er vi jo veldig fornøyd med kunden, og måten vi får jobbe på da. Det er jo et tillitsforhold uten, uten like vil jeg si. Som har gitt oss slik at vi har veldig frie rammer, og vi får på en måte jobbe innenfor drømmesituasjonen.

A-2-32-3

Det er lav terskel for å ta kontakt, slik at kunden ringe gjerne ofte og vi kan ringe uten at, det behøver ikke å være noen formaliteter rundt

A-2-33-3

vi prøver jo å understreke behovet av å prioritere oppgaver da. At vi faktisk har en backlogg som består av oppgaver som vi ønsker å utføre for dem, men, for at de skal få det de vil ha, så er det, er det viktig at de deltar, så vi prøver jo på en måte å gjør de mer bevist på det ansvarsforholdet de har i forhold til Agile da, men jeg vet ikke om at vi prøver egentlig å endre deres meninger om det. Det blir ikke, det er kanskje ikke nødvendig her.

A-2-34-3

den er veldig tilpasset etter hvordan kunden våres er da. Sånn at, dersom dem har noen ønsker, så får vi jo, så får vi jo restrukturert slik at vi møter dem ihvertfall. Så, ja. Vi har, vi har formet den slik at den er veldig effektiv i forhold til hvordan vi kan levere fortrest mulig til dem.

A-2-35-3

det er jo det staging miljøet som jeg snakket om innledningsvis da. Der legger vi jo ut ting, og det er jo klart at dersom noe må vente på eller dersom noe vi har, laget noe som er veldig bra eller veldig fint, kan du si, så engasjere det jo veldig hos kunden da

A-2-36-3

A: På brukerhistorier, altså på backloggen, at dere på en måte plukker litt basert på hvordan det passer inn med kunden?

J: Ja, hele tiden. Det er, absolutt.

A-2-37-3

Det kan ikke jeg svare på, for jeg kom inn på et senere tidspunkt enn i starten av prosjektet, og.

A-2-38-3

Altså, de er jo veldig bevist på det med arbeidsmetodikk.

A-2-39-3

Slik at, det er også en av grunnene til at de er så fornøyde.

A-2-40-3

J: Vi opererer ikke med estimater egentlig, i så stor grad, på dette prosjektet. Slik at vi gjør oppgaver, gjør dem så fort vi kan, men det blir veldig lite spurt om tall og timer. Men selvfølgelig i de situasjonene der det blir spurt om et estimat, så bruker vi jo erfaringstall og dem erfaringstallene de sier jo gjerne at ting tar litt lenger tid enn man initielt tror ihvertfall.

A: Ja, sånn worst-case.

J: Så da tar vi jo selvfølgelig å buffre det litt. Så, det er jo, jo mer mann har jobbet med lignende situasjoner jo mer nøyaktig blir jo estimatene da.

A-2-41-3

Ja, det er vell egentlig litt blanding egentlig. For kanskje er en Sprint veldig rendyrket at her er det kun en oppgave som er i fokus, og den tar alle andre ressursene. Mens, det som kanskje er mer vanlig, er at vi i hver, vi kaller det ikke Sprinter vi da, men i hver fase, eller i hver utviklingsfase så tar vi gjerne med, og jobber med hvert vårt arbeidsområdet, hvert vårt ansvarsområdet innenfor systemet, som da vil ha nødvendigvis ulike kundekontakter. Det er kanskje mer vanlig at man fordele innenfor hver Sprint alltid, fremfor alene, altså 1 og 1. Men det er jo hele tiden bytting, alt etter hvem som på en måte har ansvar for den oppgaven da.

A-2-42-3

Ja, det jo Kenneth, i stor grad. Som har kontakten med kunden. Det er han som er med i, kan du si, utforming av kravene, og spesifiseringen. Så tar han og videreformidle de oppgavene til oss da. Men igjen, etterpå så tar jo vi gjerne direkte kontakt med kunden for avklaring og videre arbeid da. Så, det variere jo, i hvilken grad at Kenneth er mellomledet eller om han ikke er det.

A-2-43-3

A: Bruker dere e-kommunikasjon, som email, telefon, chat eller videokonferanse for å kommunisere oftere og enklere med kunden?

J: Ja, alle dem.

A-2-44-3

Nei, det er vanskelig å si. Det er nok det med at vi alltid har noen oppgaver i bakhånd da. Slik at vi sikrer produktivitet, selv om kunden kanskje ikke er tilgjengelig. Slik at vi alltid har noe å gjøre, og vi klarer å levere de timene vi har forpliktet oss til å levere. Vi er jo avhengig av at vi får de tilbakemeldingene når vi trenger dem da. Så, det er jo en safe strategi for å få levert nok arbeid, men det hjelper jo ikke akkurat for det vi trenger tilbakemelding på der og da. Men, det er jo som regel ikke noe problem da.

A-2-45-3

For det er jo litt spesielt med dette prosjektet her, vil jeg si da. At kunden har gitt veldig mye ansvar til oss, i å utforme systemet. Sånn at vi står fri til å ta ganske mye avgjørelser og kan i mange tilfeller utelukke kunden i slike ting som kanskje vanligvis ville ha vært naturlig å spørre kunden om.

3 Anders, 26.02.2015

A-3-1-0

Jeg vil at vi bruker en blanding, altså vi bruker Kanban tavle og har, ja, vi bruker Scrum også. Elementer av Scrum, som er, altså vi har daglig standup møte og vi har også jevnlig leveranse til kunden også. Men, vi har ikke noen strengt definert Sprinter for eksempel. Så vi har vell en blanding mellom Scrum og Kanban vil jeg si.

A-3-2-0

Kunden og brukeren er stort sett den samme. Altså Miljødirektoratet er kunden, og Rovbase sin kunde er også, nei unnskyld, brukeren er også i Miljødirektoratet.

A-3-3-0

Foreløpig, nå har jeg bare vært her i en og en halv måned, snart to måneder nå. Jeg begynte i januar. Så jeg har foreløpig ikke opplevd noen utfordringer eller problemer med kunden sånn sett. Så, nei. Tror ikke det.

A-3-4-1

A: Men har dere opplevd at kunden kan være skeptisk til deres bruk av Agile?

E: Nei, det har jeg ikke opplevd.

A-3-5-1

Den fysiske avstanden til kunden er ikke veldig stor.

A-3-6-3

Kommunikasjon på telefon, epost går også etter mitt inntrykk, veldig bra.

A-3-7-1

Naturgårdsverket er i Sverige, og Danmark har vi jo og data for. Men, ja, det er engasjerte folk som er med i Sverige

A-3-8-1

Ja, tidvis så skjer det, det gjør jo det. Ja, det er sånn som vil alltid skje, vil jeg anta.

A-3-9-1

Miljødirektoratet og Naturgårdsverket er jo store organisasjoner sånn sett. Så det er, mitt inntrykk er at de er positivt innstilt til Agile, og, ja, på tross av sin store størrelse.

A-3-10-1

det er nok noen som vil ha mer definerte, eller de fleste vil ha konkrete tall og estimater ser jeg for meg.

A-3-11-1

Ja, det er forskjell på personene innad i bedrift, i kundene som jeg jobber for, det er det klart. Så hos enkelte så vil forståelsen være høyere for at vi ikke alltid kan gi noen nøyaktige tall eller vurderinger på jobben vi gjør. Ja, det er personavhengig.

A-3-12-2

Til en viss grad så må mann gi litt ekstra for å prøve å nå en deadline. Men, likevel, og kundeavhengig, så er det større eller mindre aksept for at vi overskriver de deadlinene, de fristene som har blitt satt. Inntrykket mitt av Miljødirektoratet som kunde, er at de, ikke er kjempestreng på akkurat det punktet. At de, det er greit for de at, om ting tar litt lenger tid. Så lenge at vi selvfølgelig leverer noe som er bra. Og det, inntrykket mitt og er at de har tillit til oss som leverandør.

A-3-13-2

Foreløpig ikke, nei. Nå har ikke jeg vært mye med på akkurat den kravspesifiseringsdelen. Stort sett så har saker og ønsker kommet inn til oss og vært rimelig bra definert på forhånd. Så, det kan virke som at kunden har en god forståelse for hva info vi trenger som leverandør for å kunne levere bra funksjonalitet, et bra produkt. Ja, de har, ja, teknisk innsikt, litt teknisk innsikt til å skjønne det.

A-3-14-2

A: Men dere har alltid visst hvilket, hva dere skal prioritere, så lenge du har vært her?

E: Ja, jeg har det.

A-3-15-2

A: Kan det være problemer knyttet til å få nok tilbakemeldinger fra kunden på deres arbeid?

E: Noen ganger, ja, det kan være at, vi legger jo ting ut i testmiljøet først og som regel så er de veldig flinke til å teste men, det går jo ikke an å rekke over absolutt alt alltid. Så, ja, men jevnt over så fungerer det veldig bra synes jeg. Ingen spesielle utfordringer akkurat der spør du meg.

A-3-16-2

Altså, dersom det, dersom enn kunne fått svar og tilbakemelding umiddelbart så hadde den vært på sitt mest effektive. Sånn er det, sånn er det ikke rett og slett, det går ikke an.

A-3-17-2

Generelt sett så er det, vil jeg tro det er lite tap av produktivitet, grunnet manglende tilbakemelding.

A-3-18-2

Nei, det er ikke det. Har ikke opplevd at Miljødirektoratet har satt seg på bakbenene for noe.

A-3-19-3

Det, siden starten av dette prosjektet der jeg kommer inn i det, så har det vært veldig innarbeidet den metodikken vi har, og det virker som det er veldig god forståelse og aksept for den hos kunden.

A-3-20-3

Nei, det er godt spørsmål, det er jeg interessert i å finne ut selv.

A-3-21-3

Ja, det tror jeg, det vil jeg si. Det er viktig å legge ting uti et testmiljø og så få kunden til å prøve det, og se hva de tenker og, ja, inspirere til mer, videre tenking.

A-3-22-3

Ja, det vil jeg si at vi gjør. Når noe kommer som er viktigere enn noe annet, selv om vi holder på med en annen sak, så er det vanlig at den saken får prioritet fremfor den andre. Det virker også som at kunden har god forståelse eller at de har et spesifikt ønsket om, ikke minst. Ja, akkurat det er ganske kundestyrte.

A-3-23-3

Ja, jeg aner ikke, nei, jeg vet ikke.

A-3-24-3

Nei, men, jeg tror vi har en ganske tydelig dialog med kunden om hvordan vi jobber også, ja, de kjenner til det.

A-3-25-3

Vi opererer ikke med akkurat et slikt begrep, slingsrom eller buffer.

A-3-26-3

Nei, jeg vil ikke si det.

A-3-27-3

Nei, jeg tror jeg skjønner hva du mener, nei det er vell ikke det. Det nærmeste, altså vår hovedkontakt med Miljødirektoratet er Kenneth. Så, men, ja, han Kay som er syk i dag, han er og, han har og litt, sånn, support, men det er ikke på det samme nivået det akkurat. Nei, så det er, det er ikke noen som fungerer som det i dette prosjektet.

4 Stian, 26.02.2015

A-4-1-0

Jeg holder på med andre prosjekter enn Rovbase, men det er i samme teamet på en måte.

A-4-2-0

SNO er kunden min.

A-4-3-0

Det blir jo som en blanding mellom Scrum og Kanban, vil jeg påstå. Vi har jo den flotte tavlen som er Kanban inspirert. I tillegg til at vi har, vi har egentlig ikke noen faste Sprinter eller noe sånn, men vi gjør jo ting stykkevis. Kunden prioritere hva vi skal gjøre, så gjennomfører vi det, og så blir det sånne runder da, så det blir jo Sprinter på en måte, men det er ikke noen faste tidsperioder.

A-4-4-0

Vi tar det litt på feelingen egentlig, ihvertfall gjør jeg det. Når jeg føler jeg trenger å dytte ut en ny versjon. For eksempel viss det er noe viktig vi har fikset, eller noe sånt. Da er det greit å få ut en versjon. Så vi tar det litt etter behov egentlig.

A-4-5-0

S: Ja, kunden min er statens naturoppsyn for felthåndboken og miljødirektoratet for hjortviltsregisteret. Og, ja, også er jo Anne Kari, hun er jo prosjektleder på felthåndbok-prosjektet. Så det er jo hun som har mest kundekontakt da, egentlig.

A: Og, brukeren, er det...

S: Brukeren er, for felthåndboken er jo det ansatte i statens naturoppsyn og eksterne ressurser som de bruker. I Hjortvilt så er det kommunen og ansatte, og egentlig alle som har noe med jakt å gjøre. Jegere, og oppsynsmenn, og ja. Folk som er ansvarlige for jaktområder, de må inn å registrere alt mulig rart.

A-4-6-1

Det vi mest sliter med er kanskje at det er vanskelig å få tak i kunden, at de er litt for opptatt til å svare på spørsmål. Det nok det vi sliter mest med, tror jeg, ihvertfall for felthåndbok.

A-4-7-1

Jeg tror ihvertfall ikke mine kunder er, sånn 100% klar over hvordan vi jobber da. At vi jobber på en spesiell metode egentlig.

A-4-8-1

S: Nei, de ligger ganske nærme da egentlig, borte på, borte med Pirbadet.

A: Ja, er det noen svensk kunde og, som du må?

S: Nei, det er det ikke. Jeg forholder meg til dem borte der.

A-4-9-1

Det kan jo føre til at ting bare hopper seg opp. Vi sitter å venter på tilbakemelding og får ikke gjort ferdig saker. Så da bare bygger det seg opp, og

da bare må mann begynne på noe nytt, for at mann ikke skal sitte å tvinne tommer, og vente på svar. Så det, ja det kan hope seg opp litt med oppgaver.

A-4-10-1

Ja, det kan jo ha det, for de kan jo ha noen retningslinjer, eller ting i anbudet sitt for eksempel, at viss du skal ha prosjektet så må du jobbe slikt og slikt. Men nå har ikke vi det da. Så da, dem er veldig greie sånt sett, kunden vi har.

A-4-11-1

Nei, det er egentlig ikke det.

A-4-12-1

Ja, det kan jo hende at, greit visst dem hadde, visst dem visste mer om hvordan vi jobbet, at de hadde vært mer hyppige på tilbakemeldinger.

A-4-13-1

Det kan jo være et problem i finansiering, mot store kunder. Der er det jo alltid, de har en liten pengesekk og sparer på alt mulig og, de må prioritere hardt for å få det de vil ha da.

A-4-14-2

Ja, jeg nevnte det jo litt i stad, at saker kan hope seg opp, dersom det er mangel på tilbakemeldinger. Og, viss pengesekken går tom så, da er det jo stop.

A-4-15-2

Ja, vi har jo nylig hatt noe sånt, vi har jo utviklet en, det er ikke en ny idé, men vi har, vi har pusset opp kanskje, en del av løsningen. Kan vi si. Og, når vi gjorde dem estimatene der, så ble det ikke lagt på en sånn buffer på timeantallet, så det var liksom nøyaktig det vi tenkte uten å legge på et buffer, så mann gjerne gjør, for å være litt sikker. Så da sprakk det jo veldig da, på timeantallet. Så det blir det litt press på å utvikle det da, kanskje.

A-4-16-2

Nei det synes jeg fungerer ganske greit. Vi, når det er noe som feiler så kommer det gjerne en mail fra kunden, og da registrerer vi saken i Gira, og så får han da etterhvert prioritere den saken sånn som han vil. Så tar vi tak i den.

A-4-17-2

Ja, det eneste problemet med det er jo egentlig bare sene tilbakemeldinger. For det, det er jo kunden som har å gjøre da.

A-4-18-2

Ja, vi har jo en slik rutine at når vi er ferdig med en sak, så legger vi ut en test, og så sier vi ifra til kunden at alt dette her er nytt på test, og da skal han gjerne gå inn og godkjenne at alt fungerer. Og det er gjerne der at det stopper opp litt da, det kan ta tid, før han setter i gang.

A-4-19-2

Ja, mann mister litt, mister litt rytmen i releasene kanskje. Når det hoper seg opp sånn. Men, vi har jo alltid noe å jobbe med, så det er ikke tapt tid sånn sett. Så, det går kanskje ikke så mye utover effektiviteten. Vi jobber jo hele tiden, så. Det er bare litt sånn hickups i, når ting skal, legges ut i produksjon, egentlig.

A-4-20-2

Nei, egentlig ikke. Jeg tror ikke vi har klart å gjort en kunde så sur at vi aldri får oppdraget.

A-4-21-3

Altså, får vi ikke tilbakemelding så er det jo bare å mase og sånt. Kommer ikke akkurat på noen andre tiltak vi gjør nei, spesifikt for å hjelpe på.

A-4-22-3

Nå har ikke jeg gjort det, kanskje mer med hun Anne Kari.

A-4-23-3

Jeg har ikke følt at det har vært noe behov egentlig.

A-4-24-3

Ja. Det blir jo ikke demoer i den forstand. Men vi har jo den testløsningen, der de selv kan gå inn og se, og alt som er nytt egentlig.

A-4-25-3

Vi har kanskje ikke hatt det målet når vi har hatt demoer.

A-4-26-3

Vi, det er mest kunden som bestemmer akkurat prioriteringen da, men det kan jo være vi tar noen saker, så skjer det, veldig fort gjort å gjøre en sak, så gjør jeg det. Viss det er greit å få gjort det liksom. Men, ja, det er mest kunden som bestemmer det.

A-4-27-3

Nei, jeg var ikke med i starten. Så det, vet jeg ikke om ble gjort.

A-4-28-3

Altså, jeg synes det har fungert greit å gjøre det sånn her egentlig. Jeg tror ikke dem er klar over at vi jobber på en sånn Scrum/Kanban måte.

A-4-29-3

Ja, vi prøver jo å gjøre det hver gang egentlig, når vi estimerer. Det er jo alltid slik når du estimerer at det tar lenger tid enn du tror. Så, mann må legge på sånn buffer. Viss ikke sprekker det.

A-4-30-3

Vi bestemme egentlig litt selv. Og kunden er jo også egentlig ganske jevnt over tilgjengelig. Så da, det er egentlig ikke noe jeg tar tenkt over, å gjort.

A-4-31-3

Kenneth har jo, det er jo han som er kundeansvarlig imot Miljødirektoratet.

A-4-32-3

Ja, på Hjortevilt så, så er det jo litt annerledes for der er det et firma som heter Naturdata som drifte mer eller mindre Hjortevilt registeret. Det er dem som gjør support og alt sånn. Så dem har jeg jo kontakt med på Skype og email. Så der går det hyppig i kommunikasjonen. I felthåndboken så er det mest epost eller møter. Synes det fungerer greit.

5 Anne Kari, 26.02.2015

A-5-1-0

Ja, som en underdel kan mann kalle det da.

A-5-2-0

A: Og hvilket Agile utviklingsmetoder bruker dere i dette prosjektet?

K: Ja, jeg vet ikke om jeg hadde satt noen på den, en forenklet variant ihvertfall, kan vi kalle det.

A-5-3-0

Vi har ikke satt noe tids-scope nei, vi tar det, vi dele opp oppgaver såpass håndterlig, og så tar vi og plukker oppgaver etterhvert, og så kommuniserer vi med kunden etter når skal faktisk ha en leveranse. Så det, vi tar det litt på sparket når det passer seg egentlig, når de har fått testet unna, og dersom de føler at nå er det på tide.

A-5-4-0

Det er oppsynsmenn i statens naturoppsyn som er sluttbrukeren

A-5-5-0

A: Og kunden er?

K: Statens naturoppsyn.

A: Ja stemmer.

K: Så det er litt sånn forvaltningen sitter sentralt i Trondheim, mens sluttbrukerene sitter ute i hele Norge da.

A-5-6-1

Ja, det er jo alltid noe utfordringer der, men de er utrolig fleksible og greie, denne kunden her da. Så det går stort sett veldig bra, det eneste er jo at dem er litt lite tilgjengelig til tider. Så vi sliter litt med å få avklaringer. Det er egentlig den største utfordringen vi har.

A-5-7-1

Det er jo det at det er en statlig kunde, og da har dem et budsjett og forholde seg til. Og dem vil gjerne ha et tall, et estimat. Det kan være litt utfordrende, for at i en sånn smidig verden, så kanskje, ja, da skal vi egentlig ikke gi et estimat på store deler av, altså, kravspek og. Så den er litt utfordring ja. Men de er ganske fleksible altså. Men når det gjelder såne store prosjekt så vil dem gjerne ha et tall å forholde seg til.

A-5-8-1

Jeg har egentlig ikke frontet at det vi gjør er Agile. Så de er med på det vi foreslår. Så jeg føler ikke at de er skeptiske. Det er jo det eneste at vi må pushe på for å få litt mer, større estimat, og slike ting da. Men utover det så, ja, de er med på det vi foreslår.

A-5-9-1

Nei, de sitter jo på Brattørkaia.

A-5-10-1

Er litt usikker.

A-5-11-1

Det er generelt ja, det er generelt at vi ofte knytter oss til et estimat, vi må gi estimat til type en løsning. Så det er litt utfordrende.

A-5-12-1

Men, akkurat med denne kunden her så synes jeg at dem er ganske dyktige på å se, komme med kravene sine. Så ikke noe spesielt nei.

A-5-13-2

Det er jo at ting blir forsinket, når vi venter på avklaringer. At vi knytter oss til et estimat som kanskje ikke er helt realistisk vise det seg. Ja, at vi bommer på et estimat er jo en utfordring.

A-5-14-2

Ja, det har vi opplevd.

A-5-15-2

Ja, det kan være vanskelig. Og som sagt, så kan det typisk kommer frem litt for sent på en måte. At vi lager det, og så ser de da at, da ser de at det ikke var sånn som tenkt.

A-5-16-2

Ja, det er typisk ting som vi venter på, for da når vi tar inn saker, så dukker det opp nye, så er det å få prioritert hva vil skal gjøre før den andre. Det er sånn typisk ting som vi kan oppleve at forsinker oss litt da. Kan hende at vi tar feil saker, feilprioritert, ja.

A-5-17-2

Det kan det jo selvfølgelig gjøre, men i den fasen så går det mer utover dem selv da. At dem ikke får lagt ut. For det er gjerne at vi legger ting i testmiljøet og så venter vi på dem. Det går jo ganske kjapt, for da forstår de at dersom de gir avpoeng på det, så får de ut produksjonen.

A-5-18-2

Ja, vi kunne absolutt vært mer effektiv. Viss vi hadde fått avklaringer litt kjappere.

A-5-19-2

K: Det kan jo føre til at vi går ledig. For at du venter på tilbakemeldinger da, da kan du ikke gjøre noe før vi får svar. Så det kan fint skje.

A: Ja, men er det noe som skjer?

K: Stort sett ikke altså. Vi har jo som regel andre saker å ta tak i, men det har hendt seg at vi må stoppe opp.

A: Men er det noe som, på en måte går på deres kappe, eller er det deres kappe?

K: Våres, det må vi jo ta.

A-5-20-3

Nei, altså, det er jo det å ha tett kommunikasjon med kunden underveis da, både på telefon og mail, og ja, ikke ta det så formelt at du må ha møter om alt, men at du kan ta det litt sånn, underveis.

A-5-21-3

Som sagt, så har ikke vi frontet så veldig mye akkurat at det er det vi er, så vi har ikke prøvd å overbevise dem om noe spesielt. Men så vil det være noe ønsker om noe estimat, på en større utviklingsjobb, så kan det være at i den forbindelse skal prøve å få løst det litt annerledes. For vi vet at vi har bommet litt på estimer før. Så da kan det hende at vi må prøve å selge inn Agile litt sånn, ja, vi får se. Men for at nå så har ikke vi gjort noe aktivt på det egentlig.

A-5-22-3

Ja det gjør vi. Ja, ja absolutt. Det er jo dem som styrer egentlig når vi kan gi leveranse, eller dem er egentlig, vi tar det litt etter hva de har behov for. Så, det gjør vi.

A-5-23-3

Vi har ikke kjørt så mye av det egentlig. Men vi har lagt ut ting i testmiljøet, men vi burde nok ha, det vell noe som vi skal gjøre til neste store løsning, at vi skal ha med demoer. At vi, ja, vi gjorde jo ikke det i foreløpet, da laget vi litt mer klar før vi bestemte. Men vi jobber jo veldig sånn brukerorientert, med å lage skisser da, og det gjorde vi forrige runde, og det gjør vi nå at vi tegner opp systemet og lage. Så sånn sett så får vi jo opp engasjementet rundt det da, det er jo en enkel demo.

A-5-24-3

Ja, slik som vi gjør nå er jo når vi først er ferdig så legger vi ut det i testmiljøet. Så det er jo det som er på en måte demoen da, men den er jo faktisk ferdig, stort sett, ja.

A-5-25-3

Vi involvere oss ikke, vi jobber ikke så tett på kunden akkurat med oppgaver for oppgaver egentlig. Da er, dem er såpass avklart så at, vi er ikke så mye inne og er, i Sprinten da.

A-5-26-3

Nei, det har vi ikke gjort så mye, nei.

A-5-27-3

Mm, det gjør vi nok.

A-5-28-3

Ja, for det er jo ofte, ofte når vi levere et tilbud, så da må vi jo levere en pris, så vi må komme med forslag til hvordan løsningen skal være, mann må tenke i det store bildet da. Men i praksis så kan mann jo egentlig prøve å dele litt mer, enn. I etterkant når du faktisk skal gjøre det ja.

A-5-29-3

Sånn usikkerhetsbuffer, på, ja det pleier vi å ha.

A-5-30-3

Bruke, ting tar som regel mer tid enn du tror i utgangspunktet, så det er lurt å ha en buffer.

A-5-31-3

Ja, nei, nei egentlig sånn som med prosjektet her så har jeg hatt kontakten med kunden helt fra dag en ja.

A-5-32-3

Ja, jeg chatter ikke så mye med han, men resten har jeg gjort ja.

6 Kay, 04.03.2015

A-6-1-0

Jeg tror ikke vi, altså vi rendyrker ikke en metode. Vi snakket litt om dette her for et par uker siden da. Vi begynte litt å lure på det der. For det liksom, det startet vell kanskje med noe sånt Scrum lignende, men såvidt jeg har forstått nå, uten at jeg er ekspert på det, så begynner vi å gå mot Kanban. For vi bruker ikke mye av de der tingene som er i Scrum, med Sprinter og retrospektiv, og ja, alt dette. Det gjør vi ikke. Men jeg kjenner ikke godt nok til Kanban egentlig da men. Men de som har greie på det sier at det er det vi, det er det det ligner mest på.

A-6-2-0

Vi prøver å release forløpende. Om det er behov for det. Og viss, ofte er det slik at kunden sier at dette haster så da er det bare å få det gjort og så release det. Det, ja, så vi har ikke disse retrospektivene eller demoene for kunden i den grad som liksom at, som er veldig spesifikk for Scrum da.

A-6-3-0

Kunden er jo Miljødirektoratet, og de er jo brukerne også. Det er jo et fagsystem for de.

A-6-4-0

Ja, så er det jo noen svenske brukere.

A-6-5-0

Jeg vet ihvertfall at Kenneth da, av og til, at han spør de da i forbindelse med medarbeidersamtaler når de skal gi tilbakemelding på oss da. Så spør han de om hva de synes rett ut. Men sender ikke ut slike spørreskjemaer eller slike ting. Men jeg vet at det er snakk om å gjøre dette nå, for Bouvet, altså hele Bouvet da.

A-6-6-1

En av utfordringene er jo at, at vi har en veldig travelt opptatt kunde. Som er, ja, han er, eller flere der da, de er veldig opptatt da. Så det er det jo både fordeler og ulemper med. Fordelene er jo at de gir oss veldig mye tillit. Så de, altså de gir oss litt sånn overordnet oppgaver, så er det opp til oss hvordan vi løser det. Men så kan det selvfølgelig være litt frustrerende av og til at du ikke får den tilbakemeldingen du trenger da.

A-6-7-1

Nei, nei, men jeg tror ikke de, vet ikke om de merker så mye om det er det ene eller andre måten vi gjør ting på. Det med at de har den taben vår med oppgavene og at de, de har verktøyet vi bruker til å registrere oppgaver og flytte oppgaver og sånt. Det er ikke de, de blir ikke så påvirket av det.

A-6-8-0

A: Men viss du sier verktøy, det er ikke den test, skal jeg godkjenne dette?

K: Nei da tenker jeg mer om oppgaveverktøy og, at vi bruker noe som heter Gira, og der ligger alle taskene og der er vell også boards, Agile boards, som vi kan flytte sant, fra "in progress" til "done" og alt dette her.

A-6-9-3

De tester jo løsningene, for vi legger vell ut på testserveren, og de tester jo løsningene, men det er alltid ting å hente der da. Vi føler jo at vi, at vi må teste de i stor grad selv da. For det blir jo mer sånn overflødig klikk test de utfører da.

A-6-10-0

I Gira så kan du velge det da, om du vil bruke Scrum eller Kanban. Så vi bruker Kanban der også.

A-6-11-0

Jeg kjenner jo igjen tavlen fra Scrum da, det gjør jeg jo da, med, du har liksom backlogg og vi opererer jo med de begrepene, med backlogg og at du må ha todo og in-progress og done, og ja. Så det er nok en god blanding ja, av Scrum og Kanban ja.

A-6-12-1

Sånn fysisk distanse så er det ikke noe problem det.

A-6-13-1

Det er litt sånn, ja, det gjelder jo mange prosjekter egentlig. De er jo fagpersoner sant, de interesserer seg på fagfeltet sitt, og så har de et IT-system for å hjelpe seg, og da er ikke de så interessert i å, de vil bare at det skal funke. Så skulle absolutt hatt mer tid ja.

A-6-14-1

Men, nei, jeg er usikker på hva jeg skal svare på der.

A-6-15-1

Jeg har vært med å skrevet en del tilbud, og det, det blir jo ofte at du ender opp, og så er det en fastpris da. Så det er, det er vanskelig ja, å få en sånn smidig metodikk inn i kontrakter.

A-6-16-1

Nå så har vi en sånn forvaltningsavtale, så den er jo ganske smidig, for den er jo bare, det er jo bare, det er en sum det, som skal brukes i løpet av så mange år. På det prosjektet. Så det er jo smidig, men det, vi har jo fått den på grunn av at vi på et eller annet tidspunkt kom inn og gjorde en god jobb. Også, utlyste de en ny, liksom, forvaltningsavtaler, og da forvalter jo du et system som er der. Så det er annerledes viss du skulle startet et helt nytt, laget noe nytt, da vil vi varte opp med en pris ja.

A-6-17-1

Nei, altså, jeg føler ikke at de stopper oss opp på noen måte egentlig. Nei, de har ofte klart for seg hva de vil ha ja. Men igjen, så er det sånn at, jeg føle ikke at de forholder seg så veldig mye til at vi bruker Agile metoder da. Så de bestiller de, og så må jo, er det vi som må, og så, og så må de prioritere da. Så det er vell mer der.

A-6-18-1

Det er det med deltakelse fra kunden. De er veldig travelt opptatt. Så de har ikke tid, og kanskje ikke så mye interesser av å sette seg inn i alle tingene heller.

A-6-19

Det er jo det at ting tar lenger tid da. Dersom du får oppgaver, og så starter du på oppgaven, og så trenger du tilbakemeldinger og tar, drar det ut i tid, da tar ting tid da. Det er, vi har det jo greit som vi har det, fordi vi har masse oppgaver da.

A-6-20-2

Så det blir jo sånn at du gjerne starter på en oppgave og så må du be om å få tilbakemelding, og så begynner du på en ny oppgave da. Fordi det tar litt tid den kommunikasjonen.

A-6-21-2

Så vi, jeg har ikke følt noe slikt press nei, til å levere altså. Det er jo, det er jo mer travelt, i perioder, men, nei det er et veldig ok prosjekt sånn sett. Det er ikke, selvfølgelig, vi har to tidsfrister i perioder da. Men det er jo ikke et fastpris prosjekt som skal være ferdig til en viss dato, da blir det jo mer press.

A-6-22-2

Ja, det kan jo det. Det er igjen dette med tilbakemelding fra kunden, at du har en sterk deltakende kunde. For nå kunden er veldig travelt opptatt med andre ting, og det vi heller ikke har det er jo at, vi har jo ikke, vi har egentlig ikke en dedikert produkteier da. Det er utfordrende da, for det at, det er jo han jeg

skulle snakket med da, så nå snakker vi jo med masse fagpersoner rundt omkring, så, så det, det savner vi vell litt da. Og det, men det er visst snakk om at det kanskje skal komme en nå. Som skal liksom holde litt i ting og ha litt overordnet ansvar.

A-6-23-2

Det kan være at en uke så haster dette veldig, men så bare kommer det ting inn fra siden og så haster den veldig og da haster ikke den fra forrige uke så mye. Vi opplever stadig vekk det ja, at det, det er liksom det siste som haster mest.

A-6-24-2

Det blir sånn som jeg sa da, at viss du sitter og, viss du har veldig mange åpne oppgaver så får du jo på en måte ikke slutført oppgavene, så blir det hengende og så, sant, du jobber med noe og du er veldig inne i det, og så må du ta en pause fra det, og begynne på noe nytt. Og så får du svar fra kunden på den oppgaven så må du tilbake igjen, og så husker du jo ikke alt, og så må du lese deg litt opp igjen på hva som var problemstillingen her. Så det er nok et problem ja.

A-6-25-2

Men de gjør nok ikke det med vilje heller da. De bare, de har ikke kapasitet så de. Vi legger ikke så mye merke til det lenger fordi vi har blitt vant til å jobbe sånn da. Men selvfølgelig, det, jeg ser jo det på mine, altså oppgavene mine, det er jo noen oppgaver det hadde vært godt å bli ferdig med da. Vi har liksom hatt de der så lenge da. For det tar så lang tid da, og hver gang, så snakker du med kunden, så får du litt nye ønsker, og så, ok, så må vi vell fikse det også, og så er det tilbakemelding og så, ja. Og så blir det nå sånn.

A-6-26-2

Nei, altså vi må jo pushe de da, slik at vi, at ikke det sklir helt ut, at vi får levert ting. Det, det blir jo litt våre oppgaver, at vi, at det må ikke skli helt ut, at du bare liksom har ti åpne oppgaver til slutt fordi du ikke får slutført noe. Så da

må vi pushe de litt, så det blir litt vår jobb da. Viss ikke vi hadde pushet de, så kunne det ført til tap av forretning ja.

A-6-27-2

Altså nå er vi heldige siden vi har en forvaltningsavtale med en ganske ok ramme, så vi fører alt, ikke på det SNO prosjektet, der er det, de har hatt noen der de har gitt faste priser. Og, ja, jeg vet ikke om det var prisen som var for lav, eller kunden som ba om for mye, men der er det tatt timer på egen kappe, ja.

A-6-28-2

På SNO har ikke vi forvaltningsavtale i samme grad, der er det mer sånn summer, altså de har litt summer. Mens på forvaltningsvtalen (Rovbase) så tar vi, så fører vi vell alt. Men det er jo tilbake til det du nevner at, viss vi ikke hadde hatt en slik avtale da, så måtte jo vi vært mer på kunden og pushet kunden mer da. For da var det jo vår lommebok der det går utover, eller Bouvet sin lommebok.

A-6-29-3

Ja, det gjør vi nok ja. Og vi oppfordrer jo de til å bruke de verktøyene vi bruker. Ja, altså vi lager jo brukere til de, så de har tilgang til våre interne systemer. Og de kan følge med på vår, boarden vår, og ja, disse tingene. Så det vil jeg si at vi gjør.

A-6-30-3

K: Ja, jeg føler ikke det, at de har så mye ønsker i forhold til Agile. Så, vi jobber jo på den måten som vi synes fungerer for oss egentlig. Og har aldri fått noe tilbakemeldinger fra de, om at de synes vi burde jobbe på en annen måte.

A: Men, kan det være at, eller, men dere tilpasser det gjerne rundt deres behov, uten at de nødvendigvis, ja.

K: Ja, det gjør vi ja, ja. Absolutt.

A-6-31-3

Ja, vi, vi utfører jo oppgavene, og så pusher vi vell ut på testserveren. Så det er jo en form for demo der da. Men vi sitter ikke å leder de gjennom det, da er det opp til de da, og logge seg inn og klikke gjennom det vi har gjort da. Det kan godt hende vi beskriver det i en epost hva som er gjort, og. Men det er sjeldent vi møtes og får kjørt ordentlig demo med de, det er det. Men det hender jo det også. At viss det, selvfølgelig viss det er litt større ting da, så må vi gjerne bort på kontorene deres og kjøre en liten demo. Men det er heller unntaket da.

A-6-32-3

Ja, ja helt klart. Jaja, helt klart. Så då, altså då har vi gjerne spørsmålene klare da, mens vi kjører demoen, så spør vi, slik at vi får den inputen. For det er jo av og til vanskelig å formulere alltid i en epost. Så det kan være greit å vise, og ja, at de får sett det på en skjerm.

A-6-33-3

Ja, det kan godt hende ja. Altså, ja, dersom det er noe som krever mye input fra de, så kan det hende at vi kan justere prioriteten på det ja. Jada, altså, ja, for vi har jo, vi har jo stor grad av frihet til å prioritere det litt vi også da. Men det er jo kunden som har det endelige ordet da, så dersom de kommer med en oppgave og sier at denne har høyeste prioriteten, så er det den som blir gjort da. Men utenfor, utenfor, liksom, alle de oppgavene som har litt mindre prioritet, der kan vi styre det litt, og se hva som er lurt og ta.

A-6-34-3

Jeg tror ikke det er så aktuelt her.

A-6-35-3

I og med at de har tilgang til systemene våre og kan se Agile boarden vår. Så vil ikke si det nei.

A-6-36-3

K: Ja, vi ville jo gjort det. Altså nå gjør ikke vi det så mye på Rovbase, men vi legger absolutt inn bufferet når vi estimerer ja. Fordi det er jo ekstremt vanskelig å estimere. Så, det gjør vi ja.

A: Nei, jeg forstår at det ikke er så aktuelt, mest på denne her, med dette prosjektet da.

K: Ja, siden vi har inne faste summer der, så er det ikke så aktuelt nei.

A-6-37-3

Vi har liksom kontakt med de fagpersonene som vi trenger for å gjøre de forskjellige oppgavene da. Så, jeg tror ikke at det er noe sånt at vi tenker på at nå skal vi jobbe med han, og nå skal vi jobbe med han. Det er sånn at vi tar oppgavene sånn at, Kenneth jobber med de svenske brukerne, eller svenske kontaktpersonen når det er behov for det, og så jobber vi med den norske når det er behov for det. Og kanskje det er en som jobber på en liten sideseksjon som skal ha en liten ting, så da jobber vi med ham.

A-6-38-3

Ja, det er jo Kenneth. Så han, ja, han sender økonomirapporter og slike ting til kunden, er liksom den som har mest kundekontakt. Selv om vi har kundekontakt vi også, men det er han som koordinerer. Det er han som styrer til en viss grad prioriteringene internt her da.

A-6-39-3

Veldig mye kommunikasjon på epost. Ja, epost, telefon. Vi bruker ikke så mye videokonferanse egentlig. Men det er sikkert fordi de er så nærme.

A3 B-Categories

In the B-categories the quotes from the interviews are translated and sorted by research question. The codes used in this appendix are the same as presented in the A-category appendix. Many of the quotes presented are used in the thesis directly.

Cause

A-4-6-1	What we struggle the most with is perhaps getting hold of the customer, that they are too busy to answer questions.
A-4-9-1	It can lead to things just piling up. We sit and wait for feedback, and cannot complete our tasks. So then it just builds up, and then you just have to start on something new, so you don't just sit there twiddling thumbs and waiting for a reply.
A-5-6-1	The only thing is that they are little available at times. So we are struggling to get clarifications. It's actually the biggest challenge we have.
A-6-13-1	They are practitioners right, they are interested in their field, and then they have an IT-system to help them. They aren't interested to... They just want it to work.
A-1-12-1	Some challenges naturally occur when we are working with two different countries. (...) Sweden is subject to many regulations because of their membership in the EU, which we do not have in Norway.
A-1-46-1	The customer can all of a sudden introduce new guidelines from the state and government, which must be complied with.
A-2-20-1	Let's say that there is a regulation, that must be upheld in these products, since it is governmental. This will include things such as compensation, compensation rates, etc. Then there is a possibility that by 1 st of November the new applications should be included.

A-2-13-1	In Rovbase being a maintenance project, then it's not the same limitations. The purchasing process, where they want a price on the product, falls away. With us, we win a contract over a three year period, which includes maintenance of the system.
A-5-7-1	Since it's a governmental customer, they have a budget to consider. So they would like to have a number, an estimate. It can be a bit challenging, because in an Agile world, then perhaps we aren't supposed to give an estimate on large parts of the requirements specification.
A-2-15-1	There are areas where the customer, or perhaps some resources at the customer, that does not have as much competence as desired. But then you just need to take some extra rounds and come to an understanding of what we need to do.
A-3-11-1	By some, the understanding is higher, that we can not always give exact figures or evaluations on the job we do.
A-4-12-1	It is possible, if they knew more about how we worked, that they would be more frequent in giving feedback.
A-2-11-1	They (Swedish customer), are very good at communicating. They are actually more accessible than the Norwegian customers. Because they, in return, are even more eager on the digital media.
A-1-23-1	The customer is split into several fragments, and everyone that is involved in the discussions and such are very good at understanding their role.
A-2-12-1	Those that the issues do not concern are outside the loop anyway. So even if the customer is large, then maybe those involved are not equally many.
A-3-9-1	The Norwegian Environment Agency and the Swedish Environmental Protection Agency are large organizations in that respect. But it is my impression that they have a positive attitude

	towards Agile in spite of their large size.
A-1-15-1	There is great enthusiasm for it (Agile).
A-2-9-1	I think it is preferable from their (customer) side, (...) they are proud of how it is operated
A-4-7-1	I think at least not my customers are 100% aware of how we work. That we are working in a special method.
A-6-7-1	Not sure if they notice if it's one way or the other we do things.

Consequence

A-4-18-2	We have a routine that when we finish something we add a test. Then we tell the customer that everything on the test is new, and that he should go in and approve that everything is working. And this is often where things stop up a little, it can take time, before he gets started.
A-5-17-2	It will negatively influence themselves (customer), that they don't get things published. Because we put things in the test environment, and then wait for them. Then it goes pretty fast, because they understand that if they give feedback, then it will be produced.
A-6-20-2	Often you start a task, and then you have to ask for feedback. Then you start on a new task, because the communication is taking some time.
A-3-13-2	Mostly issues and wants have been reasonably defined in advanced.
A-5-15-2	It is typical that it (requirements) arrives a bit too late. That we make it, and then they see that it was not the way they pictured it.

A-6-22-2	We don't really have a dedicated product owner. That is challenging because that's the guy I should be talking to. Now I just talk with lots of practitioners, so I miss having that.
A-1-34-2	Often there needs to be prioritization between Norwegian (customer) needs, Swedish (customer) needs and the common needs. Then the different countries (customers) communicate very well together about what is important.
A-6-23-2	It could be that one week something is very urgent, then something just comes in from the side-line, and then that is very urgent, and then the thing from last week isn't so urgent anymore. We experience that all the time. It is the last thing that is the most urgent.
A-1-39-2	Wrongly implemented functionality in the work area of pigeons, predators and carnivores could lead to a issued resolution for imposition of bears, which is simply wrong. Instead of the program showing 20 bears in the area, the real number could be 3, and the resolution shouldn't have been passed. So there are large values on the line, both economical and environmental values.
A-2-17-2	It might not be what the customer needs the most, that could be a consequence. That if the customer doesn't actively get involved they will be delivered functionality that may not be what they needed the most at that time.
A-2-21-2	We will always have deadline, and yes, things have a tendency to become more hectic towards such deadlines. However, I wouldn't say that we are talking about any over-committing in any way.
A-3-12-2	It is more or less accepted that we overwrite the deadlines, those deadlines that have been set. My impression on the Norwegian Environment Agency as a customer is that they are not very strict on that particular point. It is fine for them if things take a little longer time.

A-4-15-2	When we made those estimates there wasn't placed a buffer on the number of hours. It was precisely what we thought, without putting on any buffer, which you often do to be sure. So then it burst, the numbers of hours. Then there was a little pressure on the development.
A-1-37-2	We have a good and long backlog, and constantly new technology that needs to be implemented. So we never have a production or productivity issue.
A-2-28-2	Yes, if the customer is not active, then it leads to us sitting on the fence. Because then we have to wait for unnecessary feedback, and then there is more communication that wouldn't need to take place. (...) There are always new tasks to do, but it is not certain that it fits to begin with that right there and then. Hence, this leads to downtime. This leads to less productivity for us, for the customer, you could say.
A-6-24-2	Let's say you are working on something, and you are really into it, and then you have to take a break from it, and start on something new. Then you get an answer from the customer on that task, so you have to go back again. But of course, you don't remember everything, so you have to revise yourself on the problem.
A-6-27-2	We are lucky, since we have a maintenance agreement, with fairly ok conditions, so we invoice everything. Not on the SNO project however, they have given some fixed prices. I don't know if the price was too low, or if the customer asked for too much, but they have taken costs at their own expense.

Contingency

A-1-40-3	I tell everyone that is starting the team, that it is important that clarifications that are urgent are not taken over email, but should taken over the phone.
----------	--

A-2-32-3	It's a low threshold for contact, so the customer often calls, and we can call without there having to be a lot of formalities.
A-2-45-3	It is a little special with this project, I would say, that the customer has given us so much responsibility to design the system. We are free to take quite a lot of decisions, and in many cases exclude the customer in such cases where it might ordinarily have been more natural to consult with the customer.
A-1-26-3	If they <i>don't</i> come with the necessary clarifications, they won't get the functionality delivered. Then we just keep going, and they learn very fast, that if they want things delivered, they have to deliver themselves.
A-2-44-3	We always have tasks in reserve, so we ensure productivity, even though the customer may be unavailable. We always have something to do, and we are able to deliver those work hours we have committed to deliver.
A-6-31-3	We perform the task, and then we push it out on the test server. That is a form of demo I guess. However, we <i>don't sit there and guide them through it. It is up to them to login and click through what we have done. (...) It is rare that we meet and run a proper demo, but it happens. If there are bigger things, then we may have to visit their offices and run a demo. But that is more the exception.</i>
A-1-48-3	If there is a change for Norway (customer), then Sweden (customer) will of course understand that of course the resources have to be allocated there, and vica versa.
A-3-22-3	When something comes up, that is more important than anything else, even though we are right in the middle of another case, it is normal that it gets priority over the other.
A-4-29-3	We try to do that every time we estimate. It's always like that when

	you estimate, that it takes longer than you think. So you have to put on a buffer, if not it (estimates) will burst.
A-2-29-3	We flex between different projects here. So we can always re-allocate if there is any idle time in a project. We've been involved in several projects within our team, so then we just move our resources between those projects.
A-2-42-3	Kenneth, has to a large degree, contact with the customer. He participates in specifying the requirements and specification. Then he redistributes the tasks to us. Afterward though, we often take contact directly with the customer for clarifications and for further work.
A-1-53-3	We are working with functionality for foxes at the moment. This is a section that the Norwegian Environment Agency and the Swedish Environmental Protection Agency are running. Then they have included stakeholders who can participate. When this is complete we are going into functionality for wolverines, and that's when other groups will be included.
A-3-19-3	Since the start of this project, where I come into it, the method we use has been really incorporated. It seems like there is very good understanding and acceptance for it with the customer.
A-2-34-3	If they have any wishes we restructure so that we address them. We have shaped it so that it is very effective in regards to delivering as fast as possible to them.
A-1-51-3	We are always open about using Agile.
A-4-28-3	I don't think they are aware that we are working in a Scrum/Kanban way.