Decision traceability in agile software projects

Enabling alignment between changing requirements and product goals

Alice Walden

Supervisor: Ewa Braf
Examiner: Ulf Melin
Copyright

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/her own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Alice Walden
Abstract

Agile project management emphasizes flexibility and adapting to change. Embracing change often means that specified requirements get changed, removed or replaced under the course of a software project. Another consequence of the nature of agile projects is that everything that does not directly contribute to the working software gets dropped from the product lifecycle. Traceability – the ability to trace requirements back to their origins and forward to design artifacts, code, and test cases – is one such thing that may be overlooked. At the same time, traceability may be crucial to making sure that the delivered product meets the product goals. This thesis investigates the concept of decision traceability – the ability to trace decisions that relate to the evolution of a software product, as well as the fulfillment of product goals. The purpose of this thesis is to understand the importance of decision traceability in relation to product goals and changing requirements in agile software projects. For this purpose, two research questions were developed. (1) What are the challenges of achieving decision traceability in agile projects? And (2) What are important aspects of achieving decision traceability in agile projects? An interpretive qualitative case study was conducted at an IT-consultancy firm. In the case study, two of the organization’s in-house projects were observed, and six informants were interviewed. In answer to the research questions, seven challenges and six important aspects of achieving decision traceability were identified. A conclusion that can be made from the findings is that other aspects than just well-defined processes – such as team engagement, value perception, and communication – may be essential to achieving decision traceability in agile software projects.
Table of contents

1. **Introduction** ........................................................................................................................................... 11
   1.1. **Background** ........................................................................................................................................ 11
       1.1.1. Problem definition .............................................................................................................................. 12
   1.2. **Thesis statement** .................................................................................................................................. 14
       1.2.1. Purpose ................................................................................................................................................ 14
       1.2.2. Research questions ............................................................................................................................ 14
       1.2.3. Knowledge contribution and target audience ......................................................................................... 14
       1.2.4. Limitations .......................................................................................................................................... 15
   1.3. **Disposition** ......................................................................................................................................... 15

2. **Research methodology and design** ........................................................................................................ 17
   2.1. **Research philosophy** .......................................................................................................................... 17
   2.2. **Qualitative and quantitative research** .................................................................................................. 18
   2.3. **Interpretive studies in IS research** ....................................................................................................... 20
   2.4. **Ethical considerations: seven principles for interpretive IS research** ............................................... 20
       2.4.1. Application of the seven principles as ethical considerations in the present thesis ......................... 24
   2.5. **Case study research** ........................................................................................................................... 25
   2.6. **The relationship between theory and analysis** ...................................................................................... 25
   2.7. **Research design and methods of the present study** .............................................................................. 26
       2.7.1. A qualitative, interpretive case study .................................................................................................. 26
       2.7.2. Data collection .................................................................................................................................... 27
       2.7.3. Data analysis ....................................................................................................................................... 30

3. **Literature review** ...................................................................................................................................... 33
   3.1. **Traceability in software engineering** .................................................................................................... 33
   3.2. **Requirements traceability** .................................................................................................................. 34
       3.2.1. Pre-RS traceability ............................................................................................................................... 35
       3.2.2. Awareness and value-perception of traceability .................................................................................. 36
       3.2.3. The benefits of requirements traceability ............................................................................................ 36
   3.3. **Tracing decisions and design rationale** .................................................................................................. 37
   3.4. **The agile framework** ............................................................................................................................ 39
       3.4.1. The agile manifesto .............................................................................................................................. 39
       3.4.2. Achieving agility ................................................................................................................................... 39
   3.5. **Traceability in agile projects** ................................................................................................................. 41
   3.6. **Techniques and tools for decision traceability in agile projects** .......................................................... 42
       3.6.1. Inception deck ...................................................................................................................................... 42
       3.6.2. The interaction room and key project principles .................................................................................. 44
       3.6.3. JIRA & Confluence ............................................................................................................................. 45
   3.7. **Summary of literature review** ................................................................................................................ 45

4. **Empirical results** ....................................................................................................................................... 47
   4.1. **The organization** ................................................................................................................................. 47
       4.1.1. The projects ......................................................................................................................................... 47
       4.1.2. Tools for documentation and traceability ............................................................................................. 48
List of tables

Table 1: Types and number of meetings observed ................................................................. 28
Table 2: Interview informants ................................................................................................. 28
Table 3: Pre-RS traceability problems faced by providers ...................................................... 35
Table 4: Inception deck descriptions ....................................................................................... 43

List of figures

Figure 1: Philosophical assumptions and types of research..................................................... 19
Figure 2: The hermeneutic circle (my own illustration) ........................................................... 21
Figure 3: Overview of the research design ............................................................................... 27
Figure 4: Data collection and analysis timeline ...................................................................... 30
Figure 5: Two basic types of requirements traceability (Gotel & Finkelstein, 1994, p. 97) ....... 34
Figure 6: The inception deck .................................................................................................. 43
Figure 7: Challenges and important aspects of achieving decision traceability in agile projects. 68
Figure 8: Practice-related and communication-related challenges of achieving decision traceability ............................................................................................................................................... 75
Figure 9: Team engagement, project-specific, and process improvement aspects of achieving decision traceability ........................................................................................................................................... 80
1. Introduction

In this chapter, a brief background to the subject of this thesis is given (section 1.1). A problem definition is formulated in section 1.1.1, after which the thesis statement is described in section 1.2, including a description of the purpose, research questions, intended knowledge contribution and target audience, and limitations. The disposition of the remaining chapters of the thesis is given in 1.3.

1.1. Background

Agile development has become the standard approach to software development within many organizations. Today, not striving for agility in software development projects would seem irrational for most IT professionals. At the core of agile are four values that in a nutshell, capture the difference between agile and traditional project management. These values 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). Agile development is flexible and designed to handle uncertainty and changing requirements. Rasmussen (2010) expresses this in the following way:

“Things change. What was really important one week can be descoped the next. If you create a plan and follow it blindly, you won’t be able to roll with the punches when they come. That’s why when reality messes with your plan, you change your plan – not reality.”

(Rasmussen, 2010, p. 18)

The changing nature of agile projects means that agile teams face many decisions and trade-offs every day, and though the plan will most likely change in the future, at the outset of a project, we need to start with something. Rasmussen (2010) argue that without the right context or “big-picture” understanding, making the right trade-offs in an informed and balanced way becomes impossible. The starting point in forming this “big-picture” understanding that he speaks of is to ask, why are we here? By here, he refers to the initiation of a software project. Rasmussen (2010) notes that when a project team understands the why behind the software they are building, they are better able to make informed decisions, and balance conflicting forces and trade-offs. Goldkuhl and Röstlinger (2017) claim that decisions about design, content, and functionality, in what they call digital resources (for example, software products), should be governed by goals and values. They further distinguish between different types of goals, and define product goals as desirable attributes of the digital resource itself, as opposed to impact goals, which are desirable situations arising from the use of the digital resource. It is not uncommon that conflicts between different product goals arise, and that trade-offs need to be made (Goldkuhl & Röstlinger, 2017). Goldkuhl and Röstlinger (2017) propose a solution in handling these trade-offs by evaluating polarities next to each other and thereby forcing one value to be prioritized over the other. The example they use is the trade-off between availability vs. protection of information within a digital resource. Rasmussen (2010) proposes a similar method for balancing trade-offs between product goals, but also between project attributes such as scope, budget, time, and quality. In these cases, decisions need to be made regarding where the project affords flexibility. Rasmussen (2010) argues that on an agile project, the scope is the attribute that is often most suitable to being kept flexible, as budget, time, and quality tend to be more fixed.
Accepting that the scope of a project should be kept flexible has implications for how we manage the project. The scope of a software project is defined by the specified requirements for the software product to be developed (Rasmussen, 2010). Inside the scope is what we plan to build, outside is what we do not. For the initial gathering of requirements, it makes sense that requirements best be based on some form of product goals (van Lamsweerde, 2001). However, when changes are introduced in the project, and attention is directed solely towards changing the specified requirements, some risks emerge. One is that some aspects of the project goals that the initial requirements were designed to embody, get overlooked. This is easily done as many traditional requirements engineering models fail to capture the rationale behind requirements, making it hard for stakeholders to see the business value of each requirement (Ellis-braithwaite, Lock, Dawson, & Haque, 2012; Vinay, Aithal, & Adiga, 2014).

When the initial motivation for a requirement is lost, the risk is that as changes arise in the project, it becomes hard to confirm the continued alignment to product goals. To mend this problem, requirements traceability is important. Requirements traceability is described as the ability to follow the life of a requirement in both a forward and backward direction (Gotel & Finkelstein, 1994). Gotel and Finkelstein (1994) conclude that when it comes to requirements traceability, most problems are due to a lack of traceability from before any requirements get specified to after the requirements have been specified. In other words, traceability should be a concern even before any requirements have been specified, to make sure that the underlying product goals can be traced through the requirements specification phase and beyond. Achieving the kind of traceability that makes the origin of requirements transparent, enables team members to revisit the objectives behind a requirement to gain a “big-picture” understanding and ultimately make better judgments.

But with the intense focus that is put on the effectiveness of producing working software in agile development, a common outcome is that everything extrinsic – that do not directly contribute to the working code – is dropped from the product lifecycle (Palmer, 2014). Traceability is one such extrinsic value that does not directly add to the working software, but that nonetheless is considered an essential part of software development (Palmer, 2014). According to Gotel & Finkelstein (1994), discouraging the gathering of information – a common consequence of deliverable-driven cultures – is one of the problems that lead to poor requirements traceability within projects. In an environment where time is a critical factor, and focus is always on the next delivery, maintaining traceability can be challenging. But if we can understand the challenges that may be standing in the way of achieving decision traceability in agile projects, as well as what we need to consider to attain it, we will be better equipped to create successful projects from start to finish.

1.1.1. Problem definition

Due to the nature of software projects, the late introduction of new requirements or changes to initial requirements is inevitable (Book, Gruhn, & Striemer, 2016). Being agile is, therefore considered important as adapting to change is critical for success in software projects (Jayatilleke & Lai, 2018). When requirements change and delivering working software is the focus, losing sight of product goals and how they are fulfilled through new or adapted requirements becomes a risk. Many software projects fail to use traceability effectively to support product goals and requirements (Cleland-huang, 2011). Decisions and trade-offs are made continuously throughout agile projects, some of them affecting the requirements. As a result, what is delivered by the end of a project may seem very different from what was once said. And without the ability to trace decisions and their impact on requirements and product
goals, it may be hard to ensure that the end product meets the customers’ needs (Cleland-Huang, 2012).

This is a problem for developers, project managers, and customers. Treating requirements as consisting only of processes and data, without capturing the rationale for the software system, makes it hard for stakeholders to understand how the requirements fulfill product goals (Vinay et al., 2014). This can have devastating consequences when changes to requirements during a project affect the fulfillment of product goals. In these situations, it may be hard for customers to understand why the end product fails to fulfill the initial product goals. For developers and project managers, the challenge becomes to show why the changes made to the requirements during the development resulted in the failure to fulfill initial product goals.

The problem described exists for practitioners in real-life agile software projects. I would argue that this is particularly true for projects where the development of a software product takes place away from the customer, as is often the case when an IT consultancy firm is hired to handle the project and deliver a software product.

I believe that the problem described is problematic because customers and software development professionals may be blindsided by the fact that the changes that arose during development caused them to lose sight of how requirements were supposed to fulfill product goals.

The problem may depend on the very delivery-focused culture that may result from agile practices. The awareness that things will change, together with the notion that traceability requires documentation, and documentation takes time, may cause agile developers to focus on other things (Gotel & Finkelstein, 1994). Like producing working software.

Following from the problem described above, this thesis examines the challenges of traceability in the day-to-day work of agile projects. The traceability of interest in the context of this study is the kind of traceability that enables agile project members to understand and make visible the alignment of their day-to-day work to product goals. Specifically, this includes requirements traceability and traceability of decisions made during a project. Requirements traceability is of interest as requirements are essential to ensuring that the system fulfills its intended goals (Dick, Hull, & Jackson, 2017). Traceability of decisions is of interest since tracing architectural decisions has been claimed a vital contributor to supporting software engineering processes such as change impact analysis, requirements validation, safety-case construction, and long-term system maintenance (Cleland-Huang, Mirakhorli, Czauderna, & Wieloch, 2013).

In the Oxford dictionary, the word *decision* is defined as “the action or process of deciding something or of resolving a question” (“Decision,” 2019). It is with this rather broad definition that I will use the word *decision* in the present thesis. It is thus not a specific type of formal decision, made by a specific type of stakeholder, that I am referring to. I refer to any action or process of deciding on something or of resolving a question that is part of the everyday work in an agile project. Examples of this may be the decision to include or exclude a requirement in the requirements specification, or the decision of how a requirement gets implemented.

The traceability of decisions will receive special attention in this thesis. This special attention is due to two main reasons. The first reason is that decisions can be considered the driving engines for all stages of software development (Vinay et al., 2014), consequently also through the different requirements engineering activities. The second reason is that being able to trace
the rationale behind design decisions has been identified as an essential aspect of fulfilling business goals and assessing the impact of requirements changes (Tang, Jin, & Han, 2007). I further want to note that, since decisions drive all stages of software development, the requirements themselves are decisions on some higher level. The requirements specification is the result of decisions on what is to be built, and changing requirements require new decisions to be made. For this thesis, I will, therefore, use the term decision traceability to refer to the ability to trace decisions that relate to the evolution of a software product, as well as the fulfillment of product goals. This will include traceability of requirements, but also the decisions that the specified requirements are a result of, as well as design decisions. Identifying challenges of achieving decision traceability as well as important aspects to consider in striving for decision traceability is the focus of this thesis.

In this study, I’d like to handle this problem from the perspective of software developers and managers (project managers as well as department managers), while taking into consideration the broader perspective of an IT-consultancy firm that manages agile software development projects for different customers. The perspective of software developers is interesting as finding out how decision traceability is handled in practice and is experienced by practitioners will help identify the challenges encountered in the day-to-day work of developers. The perspective of managers is interesting as managers should possess a broader view of the challenges related to decision traceability as well as implementing project processes on an organizational level. Considering decision traceability in the broader context of an organization will be valuable as identifying challenges that apply to the broader perspective of an organization will help make the results of this study more widely applicable.

1.2. Thesis statement

From the problem definition stated in the previous section, the following purpose and research questions have been developed for the present thesis. With the purpose and research questions in consideration, the knowledge contribution and target audience are then discussed, as well as the limitations of the present thesis.

1.2.1. Purpose

The purpose of this thesis is to understand the importance of decision traceability in relation to product goals and changing requirements in agile software projects.

1.2.2. Research questions

1. What are the challenges of achieving decision traceability in agile projects?
2. What are important aspects of achieving decision traceability in agile projects?

1.2.3. Knowledge contribution and target audience

This thesis will contribute to the field of information systems (IS) by examining the importance of decision traceability in agile software projects. Decision traceability, as I shall call it in this thesis, puts a focus on the ongoing decision-making of agile projects. By adopting this focus, this thesis will contribute with a new perspective on traceability within agile software projects. The aspect is that of enabling the assurance of alignment between product goals and decisions made through the evolution of an agile project.

By examining the importance and challenges of achieving decision traceability within projects of an IT-consultancy firm, this thesis will contribute to the understanding of decision
traceability as a concept. By identifying challenges and important aspects of achieving decision traceability within agile projects, in the context of an IT-consultancy firm, the findings of this thesis should be of use for two main target audiences: academics and practitioners. The academics that may have use of the results are researchers and students within fields such as informatics, management, and software engineering. The practitioners that may have use of the findings may be managers, developers, or other project stakeholders in the software development industry.

The thesis may be of use for academics by giving a practical example of the challenges of decision traceability in the real-life context of an IT-consultancy firm in Sweden. For students, the thesis may also contribute to theoretical knowledge about requirements and rationale traceability. By answering the two research questions, this thesis may be of use for practitioners by highlighting critical factors of enabling alignment between product goals and decisions made through the evolution of an agile project. The challenges and important aspects identified in answering the research questions may thus serve as guidelines on what to consider when developing and implementing practices for achieving decision traceability.

1.2.4. Limitations

Traceability will in this thesis be handled in relation to product goals and changing requirements. Thus, the parts of traceability that refer to linking requirements to code, the code to tests, and other aspects of traceability that do not specifically relate to handling changing requirements in relation to product goals, are out of scope for the present thesis.

Another limitation of the present thesis is that of the perspective on decision traceability and agile projects. The customer perspective is not represented in the case study. This means that the results of the study are limited to the perspective of developers, project managers, and managers of a IT consultancy firm. The problem defined is thus studied from a supplier perspective, leaving the customer perspective out of the scope of the present thesis.

As described in the problem definition, the term decision is used rather broadly to refer to the everyday decisions of agile software projects. This means a limitation when it comes to precision of the terminology used. By this I mean that different types of decisions have not been defined or analyzed in different ways. Instead, a general understanding of decision traceability has been the focus for the present thesis.

The type of artefact being developed, and how the type of artefact affects the importance of decision traceability is not something the present thesis investigates or answers to. Instead the focus is on the project process rather than the specifics.

1.3. Disposition

In the present chapter, an introduction to the topic and motivations for the present thesis have been presented. The purpose and research questions have been defined. The next step is to discuss the methodology and research design applied for the present thesis. This is done in chapter 2, which starts with a discussion of research philosophy and ends with the methods applied in the empirical study of the present thesis. Chapter 3 consists of a literature review of areas relevant to the purpose and research questions of the present thesis. In chapter 4, the results from the empirical study are presented. In chapter 5, the results are discussed in relation to the research questions. A conclusion of the results, implications, and knowledge contribution is
given in chapter 6. Lastly, in chapter 7, my reflections on the methodology, results, and future research are discussed.
2. Research methodology and design

In this chapter, a brief introduction to research philosophy is given (section 2.1), after which the use of quantitative vs. qualitative research methods is discussed in section 2.2. In section 2.3, the use of interpretive research in IS studies is discussed along with principles such studies can be evaluated by. In section 2.4 approaches to theory and analysis are discussed along with an explanation for the approach chosen for the present study. In section 2.5 the philosophical and methodological choices for the present thesis are summarized along with a description of the methods used in the data collection and analysis.

2.1. Research philosophy

In much the same way as the product goals of a software product should guide the development of the software requirements specification, the research approach should be guided by the goals of the research. On the highest level, the goal of research in any field is to develop knowledge. Within philosophy, our assumption of what knowledge is and how it is obtained is termed epistemology (Irivari, Hirschheim, & Klein, 1998). Knowing one’s epistemological assumptions helps the researcher understand the grounds of their knowledge, an important starting point in understanding the validity, scope, and limitations of that knowledge (Myers, 2013).

Within research epistemology, a common distinction is made between positivist, interpretive, and critical research. The philosophical assumptions underlying positivism generally are that reality is objective, measurable, and independent of the observer (Myers, 2013). In positivist research, it is assumed that what is being measured is separate from the researcher and their instruments. Positivists in the social sciences apply methods and tools similar to those of the natural sciences. They aim to explain the studied phenomena in terms of independent and dependent variables, which are then used to test hypotheses (Myers, 2013).

Interpretive research assumes that knowledge of reality is socially constructed. Access to knowledge is thus assumed to depend on social constructions such as language, consciousness, shared meanings, and instruments (Myers, 2013). Therefore, interpretive researchers do not attempt to test hypotheses or pre-define independent and dependent variables. The focus is put on the complexity of human sense-making (Myers, 2013), and understanding social phenomena by the meaning that people assign to them (Klein & Myers, 1999; Myers, 2013). Dubravka and Kennan (2018) state that the purpose of research in interpretive studies is that of describing and understanding phenomena in the social world, and their meanings in context. A common assumption within social science is that the social scientist is not an observer standing outside of the subject matter looking in, but rather that they need to look at the phenomenon from the inside in order to understand it (Myers, 2013). When considering how one’s philosophical assumptions influence the research approach, it helps to examine the chosen purpose of the research and the posed research questions, as they often reflect the researcher’s assumptions about knowledge. Let’s again look at the purpose of the present thesis.

The purpose of this thesis is to understand the importance of decision traceability in relation to product goals and changing requirements in agile software projects.

The word understand is used in the purpose, and it is a keyword in considering the assumptions made on the nature of knowledge for the purposes of the present thesis. The purpose does not mention any causal relationships between variables. And it does not indicate any aspirations of hypothesis testing. Instead, the purpose is to “understand the importance of decision
traceability in relation to product goals and changing requirements in agile software projects.” This wording means that I regard decision traceability a phenomenon that needs to be understood in its context. By this I mean that decision traceability cannot simply be isolated to be studied in a controlled environment. Instead it needs to be studied in the real-life setting where it is assumed to play a role – its context. And for the present study, the context of agile software projects is the context of interest. The importance of the context, along with the idea that decision traceability is a phenomenon of the social world, was the basis for the choice of interpretivism as a philosophical foundation of the present thesis.

I regard decision traceability a social phenomenon because it exists in the sense that it is experienced by people, and should thus be understood by the meaning assigned to it. I make this claim based on the previously mentioned definition of traceability as the ability to trace. Data may be an important component of traceability, but data does not guarantee traceability. If traceability is the ability to trace, then the human actor is an important part of the tracing activity. Furthermore, the fact that traceability becomes relevant once multiple stakeholders need to understand and trace a series of developments or decisions make the social context central to traceability. This is especially true when considering the traceability of decisions, as decision-making can indeed be regarded as a social activity. Finding answers to questions of how to manage traceability in an agile project then requires understanding both the social context, where stakeholders work and interact, as well as how they experience their need and ability to trace the decisions being made.

Interpretive and positivist research are both common within the information systems and management disciplines. Critical research is much less common in these fields (Myers, 2013), but may still be considered an option. Critical research is based on the same epistemological assumptions as interpretive research. However, critical research is also based on the belief that people’s ability to change their own social and economic circumstances is constrained by various types of social, cultural, and political domination (Myers, 2013). Some interpretations of social situations are seen as preferred over others, or even imposed by one person or group upon another. An important task of critical research is to challenge the status quo by bringing to light beliefs, values, and assumptions that are taken for granted by the subjects themselves (Myers, 2013). Although critical research is based on the same philosophical assumptions as interpretivism, the choice was made to do an interpretive study and not a study within the critical research paradigm. This choice was made since the purpose, and research questions of the study were not considered to call for a critical study. This assumption is made because the research questions are aimed at identifying challenges and important aspects of achieving decision traceability and were deemed to have little to do with exposing social or cultural injustices.

2.2. Qualitative and quantitative research

Our philosophical assumptions influence the types of questions we tend to ask and guide our choice of methods when conducting research. Another way of classifying types of research besides through underlying philosophical assumptions is by the distinction between qualitative and quantitative research. This is a more common way to classify research and identifying qualitative vs. quantitative research tends to be easier, as philosophical assumptions can sometimes be hidden and not explicitly stated by the researcher (Myers, 2013). Quantitative research methods originate from the natural sciences and include survey methods, laboratory experiments, and formal and numerical methods such as mathematical modeling (Myers, 2013). Qualitative research methods, developed in the social sciences, are more focused towards understanding people, their motivations, actions, and the context of their work and lives (Myers,
Action research, case study research, and grounded theory are all qualitative research methods. The data sources used for this kind of research include different types of interviews, observation techniques, questionnaires, documents, and texts. The researcher’s impressions and reactions are also a part of the data sources in qualitative research.

Myers (2013) explains that though qualitative research is influenced by interpretive assumptions, interpretivism is not the underlying epistemology of all qualitative research. Qualitative research may be positivist, interpretive, or critical. And so is true for different qualitative research methods, which are independent of the underlying philosophical assumptions. Therefore, the selection of methods does not decide the underlying epistemology, and the choice of epistemology does not decide which methods to use. It is thus useful to return to the purpose and research questions in order to determine what type of method to use, as they are useful in different ways. Quantitative research methods are best suited if you want to have a large sample size, and you want to generalize your findings to a larger population. If the questions you are asking can be answered using statistical analysis of data, then quantitative methods are suitable (Myers, 2013). If you are interested in the social and cultural aspects of an organization, and you want to study a particular subject in depth, then qualitative methods are more suitable. As discussed in the previous section, the purpose of this thesis was aimed at understanding the importance of decision traceability in the context of agile software projects. An in-depth understanding of decision traceability as a social phenomenon was the goal rather than finding quantitative measures of its application. Therefore, the choice was made to use a qualitative research methodology.

Qualitative research methods are less applicable if you want to generalize findings to a larger population using sampling logic. However, generalization from qualitative research is possible, but it then involves generalizing findings to theory rather than to a population (Myers, 2013). Since the aim of the present thesis was not to generalize findings from a sample to a population but to gain an in-depth understanding of decision traceability in the context of agile software projects, the generalizability of qualitative methods was not considered a problem. However, the challenges of using qualitative interpretive methods in information systems research are discussed in the next section.
2.3. Interpretive studies in IS research

Interpretive research has become more important within the field of information systems (IS) research (Walsham, 2006). This can be seen as an effect of social issues related to computer-based IS being increasingly recognized (Walsham, 1995a). Myers (2017) argues that the increase of interpretive research within the IS field is partly due to a few articles published in the late 1990s that helped legitimize the use of interpretive research in IS. Two articles published by Geoff Walsham in 1995 are mentioned as important in paving the way for interpretive research within IS (Myers, 2017). In the first one, Walsham (1995b) exemplifies the application of interpretivism in IS research by highlighting the work of various IS researchers. The researchers mentioned have conducted interpretive studies on subjects such as systems design, organizational intervention of management and IS, social implications of IS, and computer-supported cooperative work and AI. The second article published by Walsham (1995a) provides guidelines for interpretive case studies in IS research. In this article, Walsham (1995a) argues that an IS researcher entering an organization is faced with complex and intertwined conceptual structures difficult to grasp without a “thick” description of what’s going on. With a “thick” description, he refers to the type of in-depth understanding that anthropological studies aim to achieve. Walsham (1995a) contributes to the legitimization of the use of interpretive case studies in the IS field by discussing the ways in which findings from interpretive case studies can be generalized. The types of generalization that he mentions are the development of concepts, generation of theory, drawing of specific implications, and contribution of rich insight.

When it comes to the applicability of interpretive research within IS, Myers (2017) argues that the technology used, and the people who use it, is always changing. The social and organizational contexts in which people live and work are changing as well. This is a challenge for IS researchers as they must understand the social and organizational contexts in which people and technology interact, not just the technology and the people (Myers, 2017). This challenge can be addressed through interpretive research. Myers (2017) argue that interpretive research is especially suitable in real-life situations that are rather messy and complicated. He recognizes that interpretive research is especially valuable when wanting to understand a new phenomenon or re-think an old problem in new ways. These arguments support the application of interpretive research for the present study as the purpose and research questions are aimed at understanding a real-life situation and identifying challenges and important aspects of achieving decision traceability in this situation.

2.4. Ethical considerations: seven principles for interpretive IS research

Another article that Myers (2017) discusses is Klein and Myers’ (1999) article in which they define a set of seven principles for evaluating interpretive studies within IS. The contribution of this article has shown to be grand as the seven principles are the most widely recognized criteria used for evaluating interpretive research (Myers, 2017). These principles are the ethical considerations I apply to ensure and evaluate the credibility and reliability of the present thesis. Let’s have a look at each one of these principles.

**Principle 1: the fundamental principle of the hermeneutic circle**

The first principle is the most fundamental one, according to Klein and Myers (1999), in that all the other principles depend on it. The hermeneutic circle refers to the notion that
all human understanding is gained by iterating between the meaning of a phenomenon as a whole, and the meaning of its parts. It suggests that understanding of a complex phenomenon is dependent on the preconceptions we form about its parts and their interrelationships. Klein and Myers (1999) use the understanding of the sentence ‘they are playing football’ as an example. Without trying to understand what is meant by the sentence as a whole, it is hard to know if the word ‘football’ refers to a round ball, an egg-shaped ball, or a ball at all. Klein and Myers (1999) describe the process of understanding the sentence as moving from an initial understanding of the individual words to the whole, and then from a ‘global’ understanding of the whole context back to a better understanding of the meaning of each word. In this example, the hermeneutic circle is a circle that starts with examining the parts, moves around to gain perspective by looking at the entire picture, and then back to the start with a new and more wholesome understanding of the individual parts and their interconnectedness.

Figure 2: The hermeneutic circle (my own illustration)

Myers (2013) illustrates how this concept relates to case study research by describing how a researcher, Sally, applies it when studying an organization. Sally starts her research by trying to gain some general knowledge of the organization (the whole). In doing so, she might look at publicly available information such as annual reports and newspaper reports (the parts). By interviewing different people in the organization, she starts to gain a better understanding of the organization as a whole, and how the different parts fit together. During her research, Sally constantly moves from the understanding of the whole, to the parts, and back to the whole.

So, the principle of the hermeneutic circle has to do with the shifting of attention back and forth between the meanings of the parts and the whole of the studied environment. This principle also highlights the interconnectedness of the data collection and analysis of an interpretive study where the researcher lets the ongoing analysis further guide the data collection.
Principle 2: the principle of contextualization
The second principle that Klein and Myers (1999) describe has to do with contextualization, referring to the social and historical background of the studied environment. In the narration of an interpretive study, the author has a responsibility to draw attention to the meaning of the context. Setting the studied phenomena in its social and historical context helps uncover how the current situation emerged. For example, a case study may be conducted where the researcher draws a conclusion that the adoption of a certain process was unsuccessful. Without giving the social and historical context of the organization and their previous processes, the findings might not be very useful. Why was it unsuccessful? What led to this outcome? If the researcher seeks to understand the how and why of current events within an organization, then the how and why of the social and historical context should also be considered, as these aspects are likely to play a big part in why a situation is what it is.

Klein and Myers (1999) further stress the fact that organizational patterns are constantly changing and that as a consequence, doing interpretive research involves trying to understand a moving target. They note as well that considering the principle of contextualization also means recognizing that in the same way as the studied organization’s history influences the results of the study, the research itself becomes part of the organization’s history. Applying the principle of contextualization furthermore requires viewing people as the producers, and not just products, of history (Klein & Myers, 1999).

Principle 3: the principle of interaction between the researcher and the subjects
Within interpretive research, the analysis springs from data that is socially constructed between the researcher and subjects. This is especially true in the case of interviews as the conversation between the interviewer and the informant is the data that is collected. To some extent, this is also the case with observation as it involves at least some type of social interaction between researcher and subjects. In light of this social nature of interpretive research, Klein and Myers (1999) underline that the participants in the study are just as able as the researcher to interpret and analyze the unfolding situations. The researcher’s presence and inquiry can thus alter the participants’ horizons by the concepts introduced by the research. For example, if the researcher asks an informant to explain how a certain process is applied in their everyday work, the informant might realize new things about how the everyday work may not be aligned with the processes that should be employed. Or the informant might even realize the process’ ineffectiveness. In this case, the data collected from the interview represents attitudes and experiences that the informant did not have prior to the interview. In the same way, a question brought up by the researcher after doing some observations might affect how the subjects view their own work.

Principle 4: the principle of abstraction and generalization
This principle regards relating the particulars of a situation, as identified by applying the principle of contextualization, to abstract categories that can have more general implications (Klein & Myers, 1999). As opposed to statistical hypotheses testing applied in positivist studies, the generalizability of concepts developed in an interpretive study depends on the plausibility and persuasiveness of the logical reasoning used in drawing conclusions from observed phenomena. Klein and Myers (1999) maintain that the application of theory is important in elevating interpretive research from being merely
Principle 5: the principle of dialogical reasoning
The principle of dialogical reasoning refers to confronting one’s prejudice and preconceptions as a researcher. As Klein and Myers (1999) assert, the intellectual basis of the research design provides a filter through which the field data is interpreted, documented, and organized. Another point to be made is that of the hermeneutic assumption that prejudice is a necessary starting point for understanding (Myers, 2013). But as researchers, we need to become aware of how our own culture and personal history shapes the way in which we view the world. Myers (2013) compares this with how, in scientific experiments, it is important to know how the research instrument is “calibrated”. Well, in interpretive research, the research instrument is the researcher, which makes it important to know how the researcher approached the research. So, dialogical reasoning refers to the dialogue between the studied phenomena and the interpreter. Though a researcher may try to be objective in the analysis, it may be hard not to inflict one’s own judgment onto the data that is collected, which will affect analysis and decisions about further data collection.

Principle 6: the principle of multiple interpretations
Much like the principle of dialogical reasoning, the principle of multiple interpretations relates to people’s preconceptions (Klein & Myers, 1999). However, in this case, Klein and Myers (1999) refer to the potentially conflicting accounts of multiple subjects. They argue that the researcher needs to consider the influences that the social context has on the actions studied and interpretations of them. In respects to this, the researcher needs to explore and document different points of view on the studied phenomena and reasons for them. Klein and Myers (1999) further argue that if the possibility of conflicting interpretations among subjects is dismissed too easily by the researcher, the reader may be left wondering why no such conflicts exist in the described case.

Principle 7: the principle of suspicion
The principle of suspicion encourages the researcher to discover “false preconceptions” (Klein & Myers, 1999). This principle is one that seems to be more applicable within the critical research paradigm as it involves considering the effects of socially created distortions and psychopathological delusions. Klein and Myers (1999) make this point and accept the possibility that some interpretive researchers may choose not to follow this principle.

Myers (2017) argues that these principles seem to have been misunderstood by many scholars. He recognizes that some seem to have used the principles as a checklist to follow in interpretive research slavishly. Instead, Myers (2017) insists, interpretive scholars should use their own judgment when it comes to the application of the principles. Myers (2017) further explains that the aim of the Klein and Myers (1999) article was to convince positivist IS researchers to accept interpretivism as a viable paradigm in IS research, an aim that seems to have been successful.
2.4.1. Application of the seven principles as ethical considerations in the present thesis

The principle of the hermeneutic circle was applied in the analysis of the empirical results by shifting the attention from interpreting the meaning of individual pieces of data to understanding the whole body of data. The way in which the data analysis was conducted to enable the application of the hermeneutic circle is discussed in section 2.7.3, data analysis. In a less explicit sense, the principle of the hermeneutic circle was applied during the data collection by intentionally focusing on trying to understand both the specifics of different findings, as well as the part they played in the whole of the studied environment.

The principle of contextualization was applied by selecting a research methodology that emphasizes the role of the context, the case study (section 2.5). In chapter 4, empirical results, the studied context is described in the first section, putting a backdrop against which the following findings can be considered. This enables and makes transparent the application of the principle of contextualization in the analysis.

The principle of interaction between the researcher and the subjects was applied in the data collection and analysis. In the data collection, this principle was applied by considering how the inquiry of the researcher might influence the subjects. How this helped shape the design of the data collection is discussed in section 2.7.2. In the analysis, the role of the researcher’s inquiry was considered in order to as accurately as possible interpret the data (section 2.7.3).

The principle of abstraction and generalization was applied in the analysis of the empirical results. The role of theory in the present thesis was part of an iterative process of data collection and analysis. This relationship between theory and analysis is further discussed in section 2.6.

The principle of dialogical reasoning was applied in the analysis of the data. The prejudice and preconceptions I, as a researcher, bring into the research process are partly based on the purpose and research questions chosen for the study. This will inevitably have affected the follow-up questions used in interviews and observations. Applying the principle of dialogical reasoning in the analysis of data means recognizing this and letting the data speak for itself to the extent possible. This was done by using quotes from interview transcripts in the empirical results chapter to make the analysis process transparent to the reader. In the discussion, explanations for the reasoning behind different interpretations are made for transparency.

The principle of multiple interpretations was applied by including participants with different roles in the data collection (section 2.7.2). Considering the different data sources (participants) in the analysis of data enabled the comparison of data with respect to the sources. This enabled potentially conflicting interpretations to be discovered. How this was done is discussed in section 2.7.3.

The principle of suspicion was not applied. This is due to the nature of the purpose and research questions of the present thesis not being deemed to have much to do with social distortions and psychopathological delusions. The purpose and research questions are of a more practical nature, making the application of this principle less relevant, in my opinion. There may, of course, be reasons for adopting the principle of suspicion in the context of the present study, but the decision was made to exclude this principle in favor of giving more attention to answering the actual research questions. That is not to say that the empirical findings were not critically assessed and analyzed – they still were. However, no specific effort to identify social distortions and psychopathological delusions was made in the analysis. Primarily since I do not deem myself as someone having the qualifications to judge psychopathological delusions in...
others. Therefore, attempting to apply this principle would have likely taken me away from the purpose of the present thesis, and without contributing much to the findings.

2.5. Case study research

A research methodology that is especially suitable when investigating contemporary phenomena in its real-life context is the case study (Myers, 2013). In case study research, the phenomenon of interest is not separated from its context, and the researcher has no control over the situations unfolding. The case study makes use of multiple sources of evidence, allowing for triangulation between different data points. Triangulation allows the researcher to gain a fuller understanding of the situation by comparing different findings on the same topic. Myers (2013) explains that case studies can be used at any stage in the research of a particular topic and that the focus of case studies is on how and why questions.

Because the case study enables a rich understanding of a phenomenon in its context, a case study became the method of choice for the present thesis. The most useful data collection technique in case studies is the interview, which serves as a window into an organization (Myers, 2013). They can help the researcher find out what people are thinking, their motivations, as well as the rationale behind their actions (Myers, 2013). In in-depth case studies, the researcher will make sure to interview many people representing diverse perspectives, but a case study can be based on only a few interviews as well (Myers, 2013). However, Myers (2013) stresses the importance of identifying ‘key’ informants that know the most about a particular topic, and who have the decision-making authority for the general area of interest. Other sources of evidence in case studies are written documents of various types, as well as different kinds of observation techniques (Walsham, 1995a). The sources of data used in the present study are described in section 2.7.2.

2.6. The relationship between theory and analysis

Another important aspect to consider in the design of the research is the relationship to theory in the analysis. Is the aim to test a theory or to build theory from empirical data? The approach where the research sets out from a general theory about the topic, that is then operationalized and tested by collecting empirical data, is called deductive reasoning (Myers, 2013). This approach is common within positivist research. When applying deductive reasoning, the outcome of the analysis is that the theory is either confirmed or not. The contrasting approach is called inductive reasoning and starts with the researcher collecting data and then letting patterns emerge in the analysis. From these patterns, a more general theory is built. Inductive reasoning is more common in qualitative research than deductive reasoning (Myers, 2013). Deductive reasoning can be described as confirmatory, and inductive reasoning as exploratory (Myers, 2013). Another approach, called abduction, is also possible. Timmermans and Tavory (2012) describe abduction reasoning as explanatory. As opposed to inductive analysis, where the researcher should put their own preconceived theoretical ideas aside during the data collection, abductive analysis encourages the researcher to enter the field with a broad theoretical base (Timmermans & Tavory, 2012). This theoretical base allows the researcher to recognize surprising findings during the research. These surprising findings will lead the researcher away from old theoretical insights into new ones. Applying abductive reasoning allows for a creative and iterative process of theory construction.

For this thesis, the approach to data collection and analysis was abductive. This choice was made in order to make use of the benefits of both deductive and inductive reasoning. The
iterative process of data collection, theory application, and analysis were considered favorable for the purpose and research questions of the study. The use of abduction allowed for interesting findings to be used as a trigger for further literature research in order to find explanations that would be useful in the analysis. The way in which the study was influenced by deductive reasoning was by reviewing the literature to form a broad understanding of topics related to decision traceability. Inductive reasoning was applied in the analysis by letting codes and themes emerge from the gathered data rather than analyzing the data using codes pre-defined as a result of the literature review.

2.7. Research design and methods of the present study

With respects to the discussions of research philosophy and approach in the previous sections, let’s now turn to the research design and methods of the present thesis. The choices regarding underlying philosophical assumptions, research methodology, and approach are summarized in section 2.7.1. Section 2.7.2 describes the methods used in the data collection, and the analysis methods used are described in section 2.7.3.

2.7.1. A qualitative, interpretive case study

As the purpose of the present thesis was aimed at understanding the importance of decision traceability in the context of agile software projects, a qualitative methodology was adopted. This choice was made as the aim was neither to test hypotheses nor to measure quantitative values. The purpose was that of understanding a complex phenomenon in context, not to confirm a general theory. This reasoning also relates to the underlying philosophical assumptions of the study. An interpretive view was adopted here. I did this as it is my belief that decision traceability is very much a phenomenon of the social world (as discussed in section 2.1).

Because I believe the social context to play an important part in the ability to trace decisions in agile projects, a case study was considered a suitable method for the purpose of the present thesis. The subject of the case study was an organization chosen as a result of convenience sampling. Collaboration with the subject organization was established prior to the definition of the thesis statement. The organization in question is an IT-consultancy firm with offices in Sundsvall and Stockholm. The case study was conducted at the Stockholm office and focused on two development projects in particular. The selection of the two projects as subjects of the case study was also a decision based on convenience as they were the only two active in-house projects at the Stockholm office at the time of the case study. The two projects were interconnected in that they were both developed for the same customer and could be regarded as two subprojects of a more general project. A more detailed description of the two projects is given in the empirical results chapter (section 4.1.1).

As mentioned in the previous chapter, the approach to data collection and analysis for the present thesis was abductive. An overview of the research design is illustrated in Figure 3.
2.7.2. Data collection

The case study evidence was gathered from three types of sources: observation, interviews, and documents. The decision to use multiple sources of data was made based on Myers’ (2013) recommendations on conducting case study research. The three data collection techniques allowed the studied phenomena to be looked at from different angles and enabled the triangulation of findings. The primary source of evidence was the interviews. Documents and observations were mostly used as tools for me as a researcher to better understand the context. This understanding contributed to the process of the study by letting me, as a researcher, better able to follow the discussions with informants during the interviews. For example, since I participated in several stand-up meetings, the informants could easily refer to a discussion from one of these meetings and not having to explain every detail of the interaction. Furthermore, I could ask specific questions that had came up as interesting considering the observations. The same goes for documents; my access to the project space on the digital channels made it possible for me to ask specifically about aspects of these channels that I didn’t understand. Thus, the use of observations and documentation may be considered mainly as supporting devices in conducting the interviews and the analysis of interview data. The deeper understanding gained from it, in my opinion, led to a more accurate analysis.

Observations

The aim of using observations in the study was to observe activities where decisions were made on how to move a project forward. These were different types of meetings that are summarized in Table 1 below. Observing and taking field notes from these activities served two purposes; (1) it gave an indication of how decisions were made and followed up on within the projects, as well as (2) gave rise to the identification of surprising evidence that then guided the further data collection and analysis. According to Walsham (2006), the advantage of the close involvement of the researcher, an effect of including observations in the data gathering, is that of in-depth access to people, issues, and data. By participating in the daily stand-up meetings of the two projects followed in the study, access was gained to the people involved. My participation furthermore helped develop trust and common ground with the team members of the projects. This may have increased their willingness to answer questions about specific events outside of the observed meetings.
Table 1: Types and number of meetings observed

<table>
<thead>
<tr>
<th>Type of meeting</th>
<th>Meetings observed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-up</td>
<td>14</td>
</tr>
<tr>
<td>Internal status meeting</td>
<td>2</td>
</tr>
<tr>
<td>External status meeting</td>
<td>2</td>
</tr>
<tr>
<td>Sprint demo</td>
<td>1</td>
</tr>
<tr>
<td>Other</td>
<td>4</td>
</tr>
</tbody>
</table>

When observing longer meetings, a computer was sometimes used for taking field notes. However, since the daily stand-up meetings were very short, no note-taking was done during these meetings. Instead, the computer was left outside of the meeting room, and when necessary, notes were taken after the end of these meetings. I made the decision to avoid taking notes during the stand-up meetings since I wanted to intrude as little as possible on the daily work of the participants. Not seeming to analyze these meetings as they took place may have helped make the team members feel more comfortable with my presence, making it easier to gain access to their interpretations and experiences.

**Semi-structured interviews**

In order to get detailed accounts of the experiences and interpretations of different stakeholders, semi-structured interviews were conducted. The objective of using semi-structured interviews was to be able to guide the focus of the discussion towards certain topics and issues that the study aimed to examine, yet allowing the respondents’ perceptions to decide what issues were put in the foreground (Myers, 2013).

Interviews were conducted with six different employees in order to collect in-depth accounts of their day-to-day work and the problems they encountered in regard to changing requirements and traceability. Four of the respondents (P1, P2, P4, and P6 in Table 2) were directly involved in one or both of the two projects. Apart from these four, two key informants in the organization were identified during the research, and interviews were conducted with these informants (informant P3 and P5) to gain a broader perspective. Selection of interview informants was thus strategic in that selecting informants with different perspectives was considered important for developing a rich understanding. Informed consent to record the interview was given orally by informants before the start of each interview.

Table 2: Interview informants

<table>
<thead>
<tr>
<th>Informant</th>
<th>Role of informant</th>
<th>Interview guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Lead developer (project Y)</td>
<td>1</td>
</tr>
<tr>
<td>P2</td>
<td>Lead developer (project X)</td>
<td>1</td>
</tr>
</tbody>
</table>
For the first two interviews, where the two lead developers (lead developer - developer in charge of a software project) of each of the two projects were interviewed, an initial interview guide was developed (Appendix 1). The interview guide included a few general questions regarding how projects are conducted within the organization and how agile practices are implemented in the projects the informants had been involved in. The remaining questions brought up change (what types of changes can during a project occur and how they are handled), traceability (what is traced in a project, what should be traced, and how are requirements handled), as well as tools (what digital and physical tools are used).

Additional interview guides were developed for the following four interviews (Appendix 2-4). The decision to use different interview guides was made primarily due to the different roles of the informants. When interviewing managers, other questions than those asked in the developer interviews were of interest. Furthermore, the use of open-ended questions in the interviews allowed the informants to bring up thoughts and ideas that they considered important on the topics discussed. Consequently, this led to the identification of new aspects of these topics to become interesting in the following data collection and analysis. For this reason, the interview guides used were reviewed before each interview. Some questions were removed once they no longer were deemed critical, in order to make time for questions that had arisen as more interesting as a result of earlier interviews. This is an example of how an abductive approach was applied in the data gathering, letting surprising evidence lead the further data collection. In the case of the last interview, the interview guide was adapted for two reasons. The first reason was that the informant was newly employed during the case study and had thus been assigned to the projects while they were ongoing. This had implications on the type of questions appropriate to ask as informant P6 had little knowledge of the history of the projects. The second reason was a recent issue in one of the projects that arose after interview P4 and surfaced as interesting in the interview with P5. A couple of questions regarding what had happened were, therefore, included to gain an additional perspective on the issue.

In terms of the role of theory in developing the interview guides, the literature review started before the interview guides were developed. However, concepts that came up and are written about in the literature review chapter were intentionally left out of the interview guides so as to not prime informants by introducing concepts they otherwise would not have brought up or considered. This decision was made as a way to account for the principle of interaction between the researcher and the subjects. The principle was applied by using open-ended questions and giving the respondents time to speak freely before asking for clarifications when needed. This technique let the interpretations of the respondents precede any interpretations by the researcher, which were only introduced.
during the interview when the responses received from the respondent seemed unclear or needed clarification. The principle of multiple interpretations was applied by selecting participants with different roles for the interviews. This enabled various points of view to be represented in the data.

**Documents**

The third source of data used was that of documents. I use the term documents broadly in this case to capture any textual data that became relevant in the case study. The documents used in this study were all some type of electronic document or other forms of digital documentation and communication. Access was given to three digital platforms used by the organization for documentation and communication: Confluence, Jira, and Slack. Confluence is an advanced wiki editor used by the projects for documentation of requirements, decisions, and other textual information about the projects. Jira is a work item tracker used by the projects to track development tasks. Slack is an application used by the teams for more informal and direct communication. The application of these tools is discussed in chapter 4. Specific documents that were part of the data collection were documents on an internal process model applied in the organization’s projects.

### 2.7.3. Data analysis

As previously mentioned, the three data collection techniques used in the case study allowed for triangulation of findings. The techniques were used in parallel during the course of the study, as illustrated by Figure 4, with an initial phase of only observation before access was granted to documents, and the first interviews were conducted. The double arrows in Figure 4 indicate that the ongoing analysis of the data from different sources helped inform and guide the data collection of other types. For example, the documents mentioned in the previous section were deemed important for the analysis as a result of an informant mentioning their existence in an interview. Another example is how some observations of the day-to-day discussions among team members led to the identifications of critical incidents that could then be referred to in subsequent interviews. This enabled the recording of more in-depth accounts of these incidents during interviews. Timmermans and Tavory (2012) note that coding and memo writing ensure thorough familiarization with data and that they, when performed against a theoretical background, are crucial parts of the abductive analysis.

![Figure 4: Data collection and analysis timeline](image)
Memos
As stated earlier, notes were taken during the research process. Some of these memos were procedural, and some were analytic. Procedural memos focus on the research process and help the researcher note what was done and how it was done in terms of the research process (Myers, 2013). Analytic memos focus on the subject matter and contain hunches and ideas about what the encountered data may mean (Myers, 2013). Writing memos during the data collection was a means of getting ideas down that might be useful in the following analysis phase. The use of memos was primarily a way for me as a researcher to collect my ideas during the process. It aided the ongoing analysis of findings, however there are no clear traces of these memos in the remaining chapters. What I mean by this is that, though the memos written during the course of the study are not presented as such in any of the following chapters, they were a tool for making sure to not forget any interpretations made by me as a researcher during the data collection. In the same way as you may take notes on ideas you want to include in a future speech, the memos served as a way of taking notes on aspects that should be interesting in the analysis. Since it was an interpretive study, the ongoing analysis during the data collection was dependent on my interpretations of unfolding events. Memos was a way to catch these interpretations so as to not lose the basis for the interpretations made. In my experience, memos were thus also a device to check the credibility of my own judgments. By putting thoughts into words, I could critically assess the reasoning behind my interpretations from a distance, asking myself “is this a reasonable interpretation”?

Coding
The semi-structured interviews were first transcribed to enable coding. Coding is a simple way of analyzing qualitative data by tagging or labeling chunks of data in order to assess units of meaning (Myers, 2013). Myers (2013) explains that coding is an analytic method that virtually reduces the size of the data and allows for the organization of the material. In the first stage of coding, descriptive coding (de Sousa, Magalhães, de Oliveira, & Albuquerque, 2019) was applied in order to identify chunks of data that conveyed similar ideas. After this type of coding had been completed, a case-by-case comparison could be performed between data from different interview transcripts. A few examples of descriptive codes that were used are requirement elicitation, requirement prioritization, customer, project initiation, change. These codes represented what the coded piece of data was essentially about.

The coding of pieces of data enabled the application of the principle of the hermeneutic circle. The codes allowed for filtering of related pieces of data which enabled comparison of the different parts of the data. The principle of the hermeneutic circle was also applied by using an additional type of coding. This was done by coding the responses from the respondents with a short description of the topic or context of the question that led to each response. In the cases that a response was chopped up in different chunks with different descriptive codes, this additional type of coding made it easy to shift from the analysis of an individual piece of data to the context in which it arose in the interview. The contextual coding also enabled the application of the principle of interaction between the researcher and the subjects. Because the questions were also coded with the same code as their respective responses, the formulation of the questions that led to each piece of data could be taken into account in the analysis so as not to misinterpret the data. This enabled transparency in the analysis as to how the words used by the researcher may have influenced the respondents.
Coding the data pieces with aliases representing the different interview respondents also enabled the application of the principle of multiple interpretations. In cases where interpretations differed, these codes helped identify which respondent said what, enabling an analysis of their different points of view in relation to their role or other aspects.

At the last stage of the analysis, when patterns had started to emerge, analytical themes and categories were abstracted from the coded data. In order to increase the readability of the empirical results chapter, the empirical results are presented according to the themes and categories identified in the analysis. Therefore, a layer of analysis is evident in the presentation of the empirical findings. The decision to present data in this way was made due to the tradeoff between total transparency and readability. In order to provide readability, and thereby making the line of reasoning of the thesis easy to follow for the reader, only data that was deemed relevant for the purpose and research questions is presented. In my opinion, this is to prefer over presenting the entire body of data, which would require the reader to perform analysis themselves in order to understand the conclusions drawn by the researcher.

**Critical incidents**

Using the critical incident approach means asking people to discuss events or incidents that the researcher deems important to the research (Myers, 2013). During the course of the study, access to both the daily stand-up meetings, and the digital documentation and communication channels, critical incidents regarding decision traceability in the projects could be identified. By observing the conversations taking place and the changes made to digital documents, a window for discussing the everyday work of the teams arose. As incidents deemed interesting occurred, they informed the further data collection using observation and interviews. For example, the issue mentioned earlier that arose during the data collection (after the interview with P4) meant that specific questions about what had happened could be asked during interviews with P5 and P6. The critical incidents handled in the study were limited to those that arose during the course of the study. In other words, no effort put into trying to analyze historical incidents within the organization or the projects.
3. Literature review

In order to understand the importance of decision traceability and how it can be achieved in agile software projects, we must first understand the concept of traceability in software engineering and the basics of agile software projects. This chapter gives a brief introduction to traceability in software engineering and the need for requirements traceability. Requirements traceability and its challenges are then discussed. Thereafter, the traceability of decisions and design rationale is discussed. The agile framework is described, and the concept of agility is discussed. Previous studies of traceability in agile projects are discussed, and a couple of agile tools are described.

3.1. Traceability in software engineering

Palmer (2014) defines traceability as a product’s highest level and most lasting structure. He further notes that

“...this is really the essence of the product and it lasts as long as the product lasts. Traceability is the access we create for ourselves to the lasting essence of our product.”

(Palmer, 2014, p. 45)

Palmer (2014) describes this essence of a product as something that exists regardless of our access to it and concludes that the wise thing to do is to build our access to this essence as we are building the product. Building this access means defining a series of external constraints on the system being built. Before this is done, Palmer (2014) argues, we do not know what we are building, and until then, we cannot consider building it effectively and efficiently. According to Palmer (2014), it is the traceability structure that becomes the product. The structure he refers to is one that links requirements, agents, functions, components, code packages, and tests together. It is what makes the product intelligible through cross-links between its parts and requirements.

Gotel, Cleland-huang, et al. (2012) present their vision for traceability in 2035 as something existing purely in the background of software development, letting project stakeholders focus on the actual work itself. In such a future, traceability would be seamless to the software engineering tasks and always exist to support the decision-making during the software life cycle. The creation and maintenance of the traceability structure would occur automatically and continuously as an effect of the use of techniques and technologies that are part of the development tasks (Gotel, Cleland-huang, et al., 2012). Gotel, Cleland-huang, et al. (2012) describe the traceability of the future as the thread that weaves data together in a project, from the rationale underlying decisions, all the way to the underlying social network that together made those decisions and is thus, best able to change them. They also note that the future of traceability is entirely requirements driven.

As of today, there are many challenges that need to be faced in order to realize this ambitious vision for software traceability. One major challenge is the tradeoff between the cost of implementation and maintenance to the benefits gained (Gotel, Huffman Hayes, et al., 2012). In the next two sections of this chapter, we will look closer at requirements traceability and traceability of decision rationale.
3.2. Requirements traceability

In a thorough analysis of the requirements traceability problem, Gotel and Finkelstein (1994) account for various aspects and problems of requirements traceability (RT), an increasingly recognized concern within the field of requirements engineering. Gotel and Finkelstein (1994) highlight the lack of a common definition of requirements traceability and identify that the various definitions used by practitioners or in the literature are either purpose-driven, meaning that RT is defined in terms of what it should do; solution-driven, and defined in terms of how it should do it; information-driven, emphasizing traceable information; or direction-driven, emphasizing traceability direction. Gotel and Finkelstein (1994) note that the definitions identified differ in their emphasis and that no one definition covers all points. They further refer to an issue of conflict between perceptions of the underlying problem of RT and conclude that “RT problem” is commonly used as an umbrella term for many problems, and that improvements within RT are expected to solve further problems. Gotel and Finkelstein (1994) suggest the following definition of RT:

“Requirements traceability refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases).”

(Gotel & Finkelstein, 1994, p. 97)

Note the choice of the word life in Gotel and Finkelstein's (1994) definition. It stresses the evolving nature of requirements. They are artifacts that change throughout the lifecycle of the product. To me, this notion speaks clearly for the need for traceability of requirements but also for the need for agility.

Gotel and Finkelstein (1994) further proposed that RT should be divided into 2 basic types: pre-RS traceability and post-RS traceability, where RS refers to the requirements specification. By pre-RS traceability, Gotel and Finkelstein (1994) refer to those aspects of a requirement’s life prior to its inclusion in the RS, and by post-RS traceability, they refer to those aspects of a requirement’s life that result from inclusion in the RS. The distinction between the two types is illustrated in Figure 5.

Figure 5: Two basic types of requirements traceability (Gotel & Finkelstein, 1994, p. 97)
3.2.1. Pre-RS traceability

Gotel and Finkelstein's (1994) findings indicate that most of the problems assigned to poor RT arise due to inadequate or a lack of pre-RS traceability. They argue that pre-RS traceability can generate improvements in quality as closed issues, as well as decisions about the requirements engineering exercise itself, can be made explicit. This would allow for decisions and closed issues from the early stage of the requirements specification activity to be re-opened and re-worked. Gotel and Finkelstein (1994) argue that this would assist auditing and repeatability. Pre-RS traceability would also provide economic leverage as it eliminates the need to reconstruct later an understanding of how the RS was produced, as is often necessary in order to use and maintain an RS in practice (Gotel & Finkelstein, 1994).

In light of the purpose of the present thesis, it seems that pre-RS traceability addresses an aspect of requirements traceability that answers to the issue of decision traceability. Pre-RS traceability regards aspects of a requirement’s life prior to the inclusion in the RS. This should thus include the source and motivations behind the requirement itself. Therefore, pre-RS traceability may enable the linking between requirements and product goals.

When it comes to the problems confronting pre-RS traceability, Gotel and Finkelstein (1994) argue that the main barrier is that the two main parties involved in the software development activity – the providers, and the end-users – have conflicting problems and needs. Gotel and Finkelstein (1994) list several problems that providers face when it comes to pre-RS traceability (listed in Table 3).

Table 3: Pre-RS traceability problems faced by providers

<table>
<thead>
<tr>
<th>Pre-RS traceability problems faced by providers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perceived as an optional extra (and of low priority), so the allocation of time, staff, and resources are often insufficient.</td>
</tr>
<tr>
<td>No allocation and management of the different roles that practitioners need to assume to: obtain and document the required information; organize it, and maintain it.</td>
</tr>
<tr>
<td>The imbalance between the work involved and the benefits gained.</td>
</tr>
<tr>
<td>Individual efforts are ad hoc and localized, whereas a combined and full-time responsibility by all is really needed.</td>
</tr>
<tr>
<td>No agreement on the end-user requirements, resulting in a tendency to focus only on their immediate and visible needs.</td>
</tr>
<tr>
<td>Concern for pre-RS traceability lessens, and concern for post-RS traceability increases, after the RS has been formally signed off. The concern must continue, but this is problematic as the activities are unpredictable, change cultures are immature, and it depends upon RT being present to do so.</td>
</tr>
<tr>
<td>Information (e.g., tacit knowledge), cannot always be obtained, and the quality of that which varies. Deliverable-driven cultures can discourage gathering Certain information.</td>
</tr>
<tr>
<td>The documentation of required information is no guarantee of its traceability. That which is structured, so it is traceable in many ways, provides no guarantee it will be up to date.</td>
</tr>
</tbody>
</table>
Gotel and Finkelstein (1994) argue that the pre-RS traceability problems cannot be addressed only through the use of technological support, due to the social nature of the activities involved. They (Gotel & Finkelstein, 1994) propose four areas of improvement to the pre-RS traceability problem: (1) Increasing awareness of information, (2) obtaining and recording information, (3) organizing and maintaining information, and (4) access and presentation of information. Indeed, several of the problems listed in Table 3 can be regarded as effects of lacking awareness of the value that pre-RS traceability can have to the development process.

### 3.2.2. Awareness and value-perception of traceability

Williams (2014) conducted a case study to examine the level of awareness and value-perception of pre-RS traceability among requirements practitioners and participants. The findings showed that the awareness of pre-RS traceability was lacking and that the need for traceability was experienced by practitioners by the end of a project. Furthermore, despite experiencing the need for traceability by the end of projects, this experience was not carried forward to subsequent projects. In other words, the lack of traceability remained within the organization’s projects even though this lack had been identified. Williams (2014) identified the major obstacle for successful pre-RS traceability to be the value perception of pre-RS traceability. Thus, improving pre-RS traceability in practice, requires addressing the value perception issue by clarifying the intentions behind pre-RS traceability.

Arkley and Riddle (2005) also highlight the lack of perceived value of requirements traceability as being the major cause for poor traceability. They found the lack of direct benefits to the main development process to be a concern for development managers and project leaders. Requirements traceability was even found to perceived as something that hindered the main development process (Arkley & Riddle, 2005). Arkley and Riddle (2005) explain that in order to alleviate development teams from the burden of requirements traceability, some projects separated their traceability processes from the main development. In these cases, a dedicated quality team, knowledgeable in traceability tools and techniques, were appointed to handle the traceability process. Arkley and Riddle (2005) report that in the cases where this separation was made between main development and traceability, the quality of the traceability information was not improved. In fact, Arkley and Riddle (2005) report that the number of incorrect traceability relationships increased, and the traceability information was out-of-sync with the development. The argument that Arkley and Riddle (2005) make is that it’s the engineers who are directly involved in the development transformation process that have the ability to record the relationships between requirements and design correctly. They are the ones who understand the reasoning behind the design decisions. This is the basis for what Arkley and Riddle (2005) call the traceability benefit problem. Simply put, the problem is that while developers are the ones who can best record traceability information, they also are the ones who do not seem to benefit in performing this task. It is this lack of perceived benefit that Arkley and Riddle (2005) reckon to be the reason for traceability tasks being assigned a very low priority by developers.

### 3.2.3. The benefits of requirements traceability

In response to the traceability benefit problem, Mäder and Egyed (2015) conducted an experimental study to find out how requirements traceability can impact developers’ performance on software maintenance tasks. They found that when exposed to an unfamiliar
software project, traceability improved developers’ performance significantly. On average developers performed 24% faster on a given task and created 50% more correct solutions with the help of traceability (Mäder & Egyed, 2015). Another study, conducted by Rempel and Mäder (2017), investigates the impact of traceability completeness on software quality. Rempel and Mäder (2017) found that the degree to which software artifacts are traceable had a statistically significant impact on the number of defects, thus improving the software quality. They further found that the impact of traceability completeness on software quality was not dependent on the size of the team. Rempel and Mäder (2017) report that with respect to the traceability use-cases high-level impact analysis, low-level impact analysis, and requirements satisfaction analysis, traceability completeness significantly improved software quality. The high-level impact analysis concerns the effect a new or changed requirement has on dependent requirements (Rempel & Mäder, 2017). It allows stakeholders to identify what requirements are impacted by a given requirement change and requires this kind of traceability links between requirements to be recorded. Low-level impact analysis concerns the effect a new or changed requirement has on dependent source code artifacts (Rempel & Mäder, 2017). Requirements satisfaction analysis concerns following the originating requirements of an implementation process to its final result to determine whether the source code satisfies all stated requirements (Rempel & Mäder, 2017).

The findings above support the claim that requirements traceability can have positive effects on developers’ performance as well as software quality. Taking these benefits into consideration may help in solving the value perception issue. If developers become aware of these benefits, they may become more invested in ensuring requirements traceability. I believe this may also help solving some of the pre-RS traceability problems identified by Gotel and Finkelstein (1994).

3.3. Tracing decisions and design rationale

The kind of traceability discussed in the previous section applies best to what is referred to as functional requirements. Functional requirements are requirements which specify what the system must do in terms of behavior; in other words, transforming inputs to outputs (Cleland-Huang & Mirakhori, 2012). Requirements that do not directly describe the intended behavior of the system are called non-functional requirements. These requirements describe qualities such as reliability, maintainability, safety, usability, portability, and security (Cleland-Huang & Mirakhori, 2012). Cleland-Huang and Mirakhori (2012) explain that non-functional requirements are significantly more difficult to trace than functional requirements. This is because they often present constraints that impact the system more broadly and are thus realized through different components and behaviors across the system architecture (Cleland-Huang & Mirakhori, 2012). In other words, the fulfillment of a non-functional requirement cannot be verified by pointing to the implementation of a specific function or behavior of the system. Instead, non-functional requirements are realized through architectural design decisions (Cleland-Huang & Mirakhori, 2012). For this reason, they have also been referred to as architecturally significant requirements (Cleland-huang et al., 2013; Mirakhori, 2011). For the purposes of achieving decision traceability in relation to product goals, I would argue that decisions having to do with the fulfillment of non-functional requirements should be just as interesting to trace as functional requirements.

The architecture of a software product is based on the requirements for the product (van der Ven, Jansen, Nijhuis, & Bosch, 2006). During the design of the architecture, many decisions and trade-offs need to be made in order to realize the requirements (Cleland-huang et al., 2013;
Mirakhorli, 2011; Tang et al., 2007; van der Ven et al., 2006). However, the justification of these decisions are often unrecorded and exist implicitly in the heads of those making the decisions (Liang, Avgeriou, & He, 2010; Tang et al., 2007; van der Ven et al., 2006).

Showing how the requirements are satisfied by the architecture, why certain design decisions were made, and how environmental conditions influence the architecture, requires understanding the design rationale (Tang et al., 2007). Design rationale captures the reasons behind design decisions and thus help architects and designers understand the reasoning behind the architecture design (Tang et al., 2007). Tang et al. (2007) argue that without traceability of design rationale, several problems may arise:

- It might be expensive to reconstruct design rationale through analysis (which would have to involve guesswork (Tang et al., 2007)).
- It might be unclear how design criteria and environmental factors influence the architecture.
- Business goals and constraints might be ignored.
- Design integrity might be violated as a result of related assumptions and constraints being omitted.
- Inaccurate assessment of the impact of changing requirements and environmental factors on the system.

van der Ven et al. (2006) mention three problems that arise when design decisions only exist in the heads of designers:

- Design decisions are cross-cutting and intertwined
- Design rules and constraints are violated
- Obsolete design decisions are not removed

That the design decisions are cross-cutting and intertwined refers to the fact that design decisions affect multiple parts of the design (van der Ven et al., 2006). When these design decisions are not explicitly expressed in the architecture, the architectural knowledge associated with the decisions is fragmented across various parts of the design (van der Ven et al., 2006). This makes it difficult to find and change decisions. The second problem van der Ven et al. (2006) mention is similar to the problem regarding design integrity that Tang et al. (2007) mention. However, van der Ven (2006) further note that violations of rules and constraints from previously taken design decisions lead to architectural drift, which is associated with problems such as increased maintenance cost. The problem with obsolete design decisions not being removed is that it tends to make the system erode more quickly (van der Ven et al., 2006), meaning that the implemented architecture diverges from the intended architecture (de Silva & Balasubramaniam, 2012). Van der Ven et al. (2006) argue that the problems mentioned above result from the focus in the software architecture design process on the resulting artifacts, rather than the decisions that lead to them.

The problems presented above suggest to me that tracing design decisions and their rationale should be an important part of ensuring decision traceability as defined for the purpose of the present thesis. Especially considering the problem regarding business goals and constraints being ignored as a result of poor traceability of design rationale. By being able to trace how design decisions fulfill business or product goals, alignment between implementations, requirements, and product goals can be confirmed.
3.4. The agile framework

“Things change. What was really important one week can be descoped the next. If you create a plan and follow it blindly, you won’t be able to roll with the punches when they come. That’s why when reality messes with your plan, you change your plan – not reality.”

(Rasmusson, 2010, p. 18)

The quote is taken from Rasmusson’s book *The Agile Samurai* (2010) in which he highlights the need for agility in software projects as requirements from the customer and the demands on the project will always change during the course of a project. The very fact that requirements will change is one of the reasons that the notion of iterative development and incremental delivery is central to agile software development (Moran, 2015).

3.4.1. The agile manifesto

The seeds of agile started already in the 1980s (Moran, 2015) and several practices such as *pair programming*, *continuous integration*, and frameworks such as *scrum* and *extreme programming* – that today fall under the agile umbrella – were being used already during the 1990s (“Agile Practices Timeline - Agile Alliance | Agile Alliance,” 2019). But it was after 17 software developers came together and created the *Manifesto for Agile Software Development* (Beck et al., 2001) in February 2001 that the agile approach to software development first found itself a solid definition that would stick. The agile manifesto takes off from four values:

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

The agile manifesto also clarifies that “… while there is value in the items on the right, we value the items on the left more”. Furthermore, the agile manifesto lists 12 principles behind these four values. The principles touch upon subjects such as teamwork, customer collaboration, and continuous delivery. They also highlight the desired attitude towards changing requirements:

> “Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage”.

(Beck et al., 2001)

3.4.2. Achieving agility

Moran (2015) notes that defining agile is harder than it seems as it is the “emergent characteristics” arising from different principles and practices that can be considered the essence of agile. In other words, agility should not be attributed to any specific techniques or rituals. Moran (2015) discusses four elements that the agile paradigm embodies; *adaptive, value-driven, collaborative, and empowered*. The *adaptive* element stresses the importance of being flexible to change and using formats that allow for adaptive planning and feedback loops. The *value-driven* element means focusing on business needs and direct assessments of progress.
Being value driven also entails tighter relationships and communication with stakeholders as well as continuous delivery of working software. The collaborative element further emphasizes the value of communication and direct interactions in exchanging information. Lastly, the empowered element involves values such as trust, respect, and courage in an environment of self-organization. Moran (2015) also notes that the traditional management role is replaced by one of servant leadership. In practice these four elements, Moran (2015) notes, often work in tandem, implying that agile is just as much a cultural stance as it is a set of practices and values.

The idea that agility relates to culture is evident in Kulak and Li’s (2017) discussion about the role of worldviews and intentions in systems thinking. They start the discussion off by referencing the so-called Noble Eightfold Path within Buddhism. In particular, they discuss four parts of this path; right view, right intention, right speech, and right action. Kulak and Li (2017) note that agile practitioners of today focus mostly on having the right speech and the right action. By right speech, they point to the focus on what to call things and to choose the right words and meanings (ScrumMaster, sprint, product owner, user story, etc.). By the strong focus on the right actions, they refer to the desire to pick the “best practices” telling them what to do. Kulak and Li (2017) argue that fixating on the right speech and action of agile is not enough and that the worldview is the foundation for the right intention, speech, and action, each of which is supported by the previous one in the mentioned order.

Kulak and Li (2017) acknowledge that each and every one of us carries our own worldview with us that we cannot easily switch off. And that instead of “fixing” each other’s worldviews to fit a project, we need to learn how to work with them. However, Kulak and Li (2017) make a seemingly contradictory claim when they go on to discuss group worldviews. They explain that, after working together for a few years, the entire staff of an organization may share a common worldview of how to achieve success. They further note that the common worldview shared by the team may limit rather than enable success (Kulak & Li, 2017). This is where Kulak and Li’s (2017) argument takes a turn, and they talk about forming group worldviews. When it comes to forming a group worldview, they regard the agile manifesto a good example of a worldview to aspire towards. How one can be successful in forming a specific group worldview, and how this compares to changing one’s own worldview, Kulak and Li (2017) are not specifically clear on. But what they are clear on is their critique against most agile coaches and consultants trying to translate the agile manifesto directly and immediately into practices. Instead, Kulak and Li (2017) propose letting the agile values form our worldview. This may be a crucial aspect in achieving agility as Kulak and Li (2017) note that we may follow the “actions” of agile or Scrum perfectly and still manage to mess things up. One reason for this, they note, is that team members may have worldviews that conflict or prevent the work from getting done.

Kulak and Li (2017) further state that another reason for projects failing despite following the agile practices is a case of incompatible intentions. Incompatible intentions for the practices used in an agile team can render the practices ineffective. The example Kulak and Li (2017) mention is the intention behind the daily standup. If a manager has the intention to use the daily standup as a means of controlling the team, they will most likely fail at the practice. Kulak and Li (2017) recognize that intentions are tricky, hard to uncover, and difficult to change, but note that they best not be viewed upon as “good” or “bad” but rather as compatible with success or not.

When it comes to using the right speech, Kulak and Li (2017) highlight a potential problem with just relying on the right words for processes, activities, and roles in agile projects. They
recognize that words carry great meaning and that choosing better words can help in changing behavior. However, they argue that changing the terminology used within an organization can be deceiving if no real changes are implemented. In such a case, the change in terminology can end up having a negative impact, reminding people of the failure to live up to the promise of change. The issue that Kulak and Li (2017) raise serves as a warning to agile teams to consider the meanings they assign to the words they use and how they want to live up to them.

In contrast to many authors of agile practice books Kulak and Li (2017) steer the attention away from “best practices” or the right actions, as they call it, and instead advise approaching the issue of agility from the other end, starting with the right worldview and intentions. They stress the importance of having a worldview that is more consistent with speed-to-market, setting intentions for creating a team that produces fewer defects, and lastly speaking in a way that does not encourage mechanical behavior.

3.5. Traceability in agile projects

Cleland-Huang (2012) discusses traceability in the context of agile projects and conclude that although the means of achieving traceability in agile projects differ from those of traditional projects, the goals of traceability are much the same. Palmer (2014) even argues that forgetting traceability in the name of agility is a dangerous trend and that traceability should be regarded as an essential feature. One that will increase our ability to realize the benefits of agility in systems development. Antonino, Keuler, Germann, and Cronauer (2014) stress the importance for each agile team to have “a consistent global view of the system being developed” (Antonino et al., 2014, p. 221) and to know why a certain design decision was taken as well as the impacts of such a decision on other parts of the system. Cleland-Huang (2012) lists six reasons for tracing in agile projects:

- **Change impact analysis** – to assess how a proposed change will impact the existing system, in order to accommodate tasks such as communication, team coordination, and effort estimation.
- **Product conformance** – to ensure that the delivered product meets the customers’ needs, i.e. realizes their requirements. This is commonly referred to as requirements validation.
- **Process compliance** – to ensure that any procedural processes, such as reviews and tests, have been conducted.
- **Project accountability** – to provide assurance that the solution does not include gold-plating (i.e., excess functionality) and that all changes match a requested feature request.
- **Baseline reproducibility** – to support configuration of baselines, so that different versions can be reproduced.
- **Organizational learning** – to document rationales behind critical decisions in order to transfer knowledge to new team members.

In a study on implementing traceability in agile projects, Jacobsson (2009) identifies potential problems with implementing traceability practices in agile projects. Some of the identified problems that may cause a lack in traceability are lack of motivation, lack of knowledge, cost of implementing traceability, and administrative overhead.
3.6. Techniques and tools for decision traceability in agile projects

In an empirical study analyzing data from 16 software development organizations, Cao and Ramesh (2008) identified seven agile requirements engineering practices. These were (1) face-to-face communication, (2) iterative requirements engineering, (3) extreme prioritization, (4) constant planning, (5) prototyping, (6) test-driven development, and (7) reviews and tests. What is noteworthy is that all these practices enable and support iterative requirements elicitation and refinement, a core aspect of agile development. However, in and of themselves, all but one of them do not provide support for traceability. Test-driven development is the one practice mentioned by Cao and Ramesh (2008) that provides some sort of traceability structure. And in this case, the traceability is from tests to the source code. The seven agile requirements engineering practices identified by Cao and Ramesh (2008) enable agility in that decisions, changes, and prioritizations can be made iteratively and continuously. But for those concerned with tracing the development process from product goals all the way to the end-product, additional practices seem to be required. Furthermore, Cao and Ramesh (2008) state that almost all of the organizations studied report inability to gain access to the customer or obtaining a consensus among various customer groups as the most common challenge when it comes to requirements engineering. In my opinion, this could be another indication that practices supporting requirements traceability in agile projects are needed.

3.6.1. Inception deck

Some of the reasons that may trigger changes during a project are related to the project vision (Jayatilleke & Lai, 2018). It may be that the problem space becomes better understood from a customer point of view, or simply that the project vision becomes clearer, resulting in changes being introduced. Other events that may trigger requirements changes can be related to the requirements specifications themselves. It may be the resolution of misunderstandings and miscommunication, or incorrect identifications of requirements that trigger changes (Jayatilleke & Lai, 2018). These triggers of change may arise due to what Rasmusson (2010) identify as a deadly problem for projects: when people have different ideas of what success looks like, and they fail to realize it.

“The assumption of consensus where none exists is what kills most projects.”

(Rasmusson, 2010, p. 49)

Rasmusson (2010) argues that the reason that many projects get killed before they even get out of the starting blocks is that they fail to ask the right questions at the beginning of a project and that they lack the courage to ask the tough ones. To tackle this problem, Rasmusson (2010) suggests employing a method called the inception deck. He refers to the inception deck as “ten questions you’d be crazy not to ask before starting any software project” (Rasmusson, 2010, p. 48). The goal of the inception deck is to get everyone on board before the project gets started, minimizing the risk of misunderstandings that can be devastating for project success. The inception deck should help the team make intelligent decisions by communicating the goals, vision, and context of the project. It should also help the stakeholders make decisions on whether or not to proceed with the project by giving them the information they need (Rasmusson, 2010). Rasmusson (2010) argues that agile methods like extreme programming (XP) and scrum do not sufficiently cover the project chartering. The inception deck can, therefore, be used as a complement and is “a fast, lightweight way to distill a project to its very core and communicate that shared understanding to the greater team and community.”
The inception deck is, according to Rasmusson (2010), about getting the right people in the room and asking them the right questions in order to set the right expectations about the project. It should be seen as a collective exercise of defining what the project is, what it isn’t, and what it’s going to take to deliver, as well as capturing the output of the activity (Rasmusson, 2010). Rasmusson (2010) argues that the people involved in the activity of producing the inception deck should be anyone directly involved in the project. Including customers, stakeholders, team members, developers, testers, analysts, and anyone else who can contribute to the effective execution of the project.

The right questions to ask stakeholders, according to Rasmusson (2010), is summarized in Figure 6, which represents the ten items of the inception deck. The short descriptions of these items, as presented by Rassmusson (2010), are listed in Table 4.

Table 4: Inception deck descriptions

<table>
<thead>
<tr>
<th>Why are we here?</th>
<th>This is a quick reminder about why we are here, who our customers are, and why we decided to do this project in the first place.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an elevator pitch</td>
<td>If we had thirty seconds and two sentences to describe our project, what would we say?</td>
</tr>
<tr>
<td>Design a product box</td>
<td>If we were flipping through a magazine and we saw an advertisement for our product or service, what would it say, and, more importantly, would we buy it?</td>
</tr>
<tr>
<td>Create a NOT list</td>
<td>It’s pretty clear what we want to do on this project. Let’s be even clearer and show what we are not doing</td>
</tr>
<tr>
<td>Meet your neighbors</td>
<td>Our project community is always bigger than we think. Why don’t we invite them over for coffee and introduce ourselves?</td>
</tr>
<tr>
<td>Show the solution</td>
<td>Let’s draw the high-level blueprints of the technical architecture to make sure we are all thinking of the same thing.</td>
</tr>
</tbody>
</table>
Ask what keeps us up all night

Some of the things that happen on projects are downright scary. But talking about them, and what we can do to avoid them, can make them less scary.

Size it up

Is this thing a three-, six-, or nine-month project?

Be clear on what’s going to give

Projects have levers like time, scope, budget, and quality. What’s most and least important for this project at this time?

Show what it’s going to take

How long is it going to take? How much will it cost? And what kind of team are we going to need to pull this off?

The inception deck can take a couple of days to about two weeks to build. Rasmusson (2010) stresses that the inception deck is a living, breathing artifact, and not something to produce once and file away. It should be revisited anytime the project is subject to a major change in spirit or direction. Rasmusson (2010) notes that it may be put on the wall in the team’s work area to serve as a reminder of what they are working on and why.

3.6.2. The interaction room and key project principles

Book et al. (2016) also stress the importance of getting all the right people in the same room for the discussion, including developers, technology, and business experts, as well as future users. They argue that this is necessary in order to make sure that the correct software is being developed rather than just that the software is correctly developed, which developers themselves can get done. To do this, Book et al. (2016) suggest using a technique they call the interaction room (IR). The interaction room is supposed to help developers understand the objectives and priorities of future users. The interaction room is a physical room with its walls covered with models of business processes, business objects, user journeys, and system landscapes (Book et al., 2016). Book et al. (2016) explain that its finite walls are what encourage stakeholders to focus on what is most important. Book et al. (2016) explain that the IR promotes targeted and moderated communication between project stakeholders. It helps participants focus on what’s important and makes sure that the required features are evaluated and prioritized in light of the desired added value (Book et al., 2016). The IR technique supports both the scoping of projects as well as the pursuit of project progress (Book et al., 2016).

Book et al. (2016) maintain that the key principles of every project are abstraction, value orientation, communication, and transparency. And that the Interaction Room ensures that these principles become tangible and visible.

Abstraction. The principle of abstraction means focusing on key relationships and genuinely essential decisions (Book et al., 2016). Book et al. (2016) explain that though this means leaving out details at certain levels of abstraction, one needs to be aware that details will need to be filled out at a later point in time. One should also be aware that the details left to be handled later may play an extremely important role in the future (Book et al., 2016). The principle of abstraction is manifested in the Interaction Room by the finite space available, reminding stakeholders to focus on what’s really important.

Value orientation. The principle of value orientation means that the decision whether a feature is required should be included, and how much effort should be spent on its implementation, should be guided by the value creation of that feature within the business model supported by
the software (Book et al., 2016). Book et al. (2016) explain that value orientation is expressed in the IR by use of model annotations that let stakeholders explicitly highlight particularly important features and dimensions.

**Communication.** The principle of communication highlights that all stakeholder groups should be represented and involved in the creation of relevant specifications and decisions. The IR makes this possible by acting as a central communication point, ensuring that communication takes place face-to-face (Book et al., 2016). Book et al. (2016) explain that the IR is used in any situation that merits actual discussion and for everything that in written communication would remain a volatile unspoken perception rather than an explicit statement. It can thus be used to negotiate and re-negotiate priorities, assess effects of late requirements, and to exchange early and late requirements (Book et al., 2016).

**Transparency.** The principle of transparency means that preliminary or final specifications and decisions are made accessible to all relevant stakeholders (Book et al., 2016). Book et al. (2016) argue that stakeholders only remain committed based on the principle of transparency and that this is important if they are to understand, support, and interpret decisions appropriately during their implementation. The IR manifests the principle of transparency by displaying the current state of the project at all times and by representing the central orientation point (Book et al., 2016).

### 3.6.3. JIRA & Confluence

Filion, Daviot, Bel, and Gagnon (2017) explain that Requirement Management tools can be costly for Small and Medium-Sized Enterprises (SME), and propose an affordable solution based on the Atlassian technologies JIRA and Confluence. Filion et al. (2017) remark that, although JIRA does not target requirements management, it can be configured to support requirements traceability. JIRA, a tool commonly used for agile projects, is a generic work item tracker that allows for tracking of software bugs and tasks (Filion et al., 2017). An advantage of using JIRA is its configurability, allowing users to create all types of work items, and linking them together (Filion et al., 2017). A shortcoming that Filion et al. (2017) mention is the lack of a built-in solution for graphical representations such as diagrams or traceability matrices.

Confluence is described by Filion et al. (2017) as an advanced wiki editor that compensates for JIRA’s inability to support textual descriptions of requirements in a proper way. Confluence can be connected to the JIRA database, which provides a way to reflect the state of requirements dynamically.

### 3.7. Summary of literature review

In this chapter I have described the concept of traceability in software engineering. Traceability was defined, by the help of Palmer (2014), as the access we create for ourselves to the lasting essence of our product. Requirements traceability was described and the distinction between pre-RS traceability and post-RS traceability was presented. In considering the purpose of the present thesis, pre-RS traceability was identified as an important part of decision traceability. The problem of awareness and value-perception of traceability was discussed, as well as the benefits of requirements traceability. The need for tracing design decisions and their rationale, as well as the problems that may arise due to the lack of traceability of design rationale was discussed. The agile framework was described and the idea of achieving agility using the right view and intention was considered. Previous research of traceability in agile projects was
presented. Lastly, a few tools that were identified as relevant in the discussion of decision traceability in agile projects were described.
4. Empirical results

*In this chapter, the empirical results of the case study are presented. In the first section, the organization in which the case study was conducted is described as well as the two projects that the case study specifically focused on. This will provide a context to the findings presented in the following sections of the chapter, and thereby enable the application of the principle of contextualization. In the second section, findings related to decision traceability are presented.*

4.1. The organization

The organization where the case study took place is an IT-consulting firm with about 80 employees in two Swedish cities. The firm specializes in development and life cycle management of digital solutions within areas such as AWS cloud, mobility, omnichannel, IoT, media, and biometrics. Their customers are based in the Nordics and come from the public sector, industry, retail, and ICT. For the sake of simplicity, the organization will be referred to as CompanyX in this chapter.

On CompanyX’s website, it is written about their aim to produce creative solutions with concrete business advantages for its customers. The application of agile methods is described as a given, with the assurance of always looking far beyond each individual sprint.

The case study took place at the smaller one of CompanyX’s two locations. In total, 15 employees worked at the studied location at the time of the case study. 13 of these were consultants; the other two were the CEO and one of the department managers. Eight of the 13 consultants in the studied location were in-house developers, meaning that they worked from CompanyX’s office rather than at a customer site. At the studied location, two main development projects took place during the case study. These two projects were the ones the case study specifically focused on and are described in section 4.1.1.

4.1.1. The projects

The two main projects that participated in the case study were both part of the same initiative from a customer in the telecom industry. The two projects could thus be considered subprojects of a single project. In one respect they may very well be considered one project as the daily standup meetings as well as a couple of other regular meetings were held together, with developers from both projects. Apart from having the same customer, the two projects also had the same project manager. The project manager was not based in the same location as the developers who were all based in the same location, working from the consulting organization’s own office. The customer was based in the same city as the projects, but as the projects were developed in-house at CompanyX’s office, the customer was not present for the everyday work of the team. Because the project manager and the customer were based offsite from the development team, the regular meetings were held using conference calls via Slack.

At the beginning of the case study, the two projects (from here on referred to as ProjectX and ProjectY) consisted of five team members apart from the project manager and the customer. ProjectX had been going on for a longer period of time than ProjectY, which was a more recently started project. The goal of ProjectX was to develop a service that other applications (such as web applications) can use to charge customers using different payment methods. The goal of ProjectY was to develop a system for coordinating different services and microservices.
ProjectX had three developers involved, and ProjectY had two at the beginning of the data collection. However, towards the end of the data collection, the lead developer from ProjectX started to get more involved in the younger project (ProjectY) and was involved in both at the end of the data collection phase.

The developers in the two projects had one to three years of experience and were described by the department manager (P5) as young and still in need of support from more senior coworkers. The projects, therefore, had an external solution architect appointed to the team, whose role was to support and coach the developers to be more effective. This coach was, like the project manager, based in Sundsvall, but came down to the Stockholm office on a regular basis. During the course of the case study, a new employee (P6) joined the Stockholm office and the two projects as an onsite project manager.

Meetings
A few different meetings were observed apart from the daily standup meetings. Two of the types were regular so-called internal and external status meetings. The internal status meetings were held with the team and worked as a briefing before the external status meeting, which also included the customer. The status meetings were weekly, and other things regarding the project status than what the daily standup covered could be discussed. The status meetings and the daily standups were held via slack conference call. The project manager and the customer participated in the conference call from their respective locations, while the team sat together in a conference room at the Stockholm office with the conference call on speaker. During the standup meetings, the daily status of each project was discussed briefly, with more focus the work within each project rather than on the individual developers within the projects. In other words, the developers were grouped together by project in a way that made the projects themselves seem to be the primary participants in the standup. The customer joined in on the standup meetings when they were available.

A few meetings were observed that took place only in a physical meeting room, meaning that all participants were physically present, and no conference call was needed. They were a sprint planning meeting, a sprint demo for the customer, and a couple of discussion meetings with the external solution architect about technical implementations. During these meetings, the customer or the external solution architect came to the office, and the meetings were held in one of the conference rooms. The conference rooms had at least one whiteboard and either a projector or a smaller monitor where digital documents could be displayed. If a whiteboard was used for drawing graphs or other visual representations during the meeting, the whiteboard was cleaned after the end of the meeting. If what had been drawn on the whiteboard was deemed valuable to save by the participants of the meeting, someone would take a picture of the whiteboard for future reference. During one of the observed meetings, a picture of a whiteboard representation from a previous meeting was used to remember what had been discussed at a previous stage.

4.1.2. Tools for documentation and traceability
The tools used in the products that become interesting in light of the purpose and research questions for the present thesis are the software tools used for documentation of requirements, features, and decisions. For the documentation of features, decisions, and high-level requirements, the projects made use of Confluence. Jira was used for the everyday task
management of the teams. In Jira, the requirements were broken down into subtasks by the developers.

The fact that Jira and Confluence are used in the organization and provide ways of linking entries of different kinds to each other, suggest great opportunities for implementing custom-made traceability practices. The ability to link entities together and define dependencies was mentioned by P4 and P2 as benefits with the tools. However, no account was given about a specific practice or pattern in which this was usually done.

4.1.3. Process model

During the data collection, it was revealed that documentation describing the processes used in the organization’s projects had been developed recently by a few managers within the organization. The process model had been documented only a few months prior to the data collection. Two of the informants interviewed, P3 and P5, had both been involved in developing the documentation. Though the documentation was new at the time of the case study, the processes the documentation describes had been applied in CompanyX for roughly five years, according to P5. However, P5 noted that the way of working that the documentation describes had only been known to a small group of people.

“they [process model documents] are rather new. They have existed- we have been working since 2014, I would say, according to that process. So, it has been known, to a small group of people, how we work, and it has been implemented also... but it just hasn’t been written down. So, it is a rather new- it is pretty recent that we actually have gotten it in print too. So, it has also been refined, of course.”

P5

The process model was based on a model called Disciplined Agile Delivery (DAD), a model that, according to P3, had been applied for the past few years. In the latest version of the model, that had recently been documented, P3 explained that an additional layer called Disciplined DevOps had been added.

Like P5, P3 noted that everyone didn’t know about the process model and explained that the ownership of the model is not clearly defined, resulting in a lack of communication about it.

“well, it really is the ownership, I would say, that is a bit fluid around this model. I was involved in developing it, but then I should probably not be the owner of it. It should be- we have like an action point that we should- or that we have to appoint someone that should own these and drive them- It shouldn’t require a lot of management, but still, someone has to feel a certain responsibility for it, and spread the message about it. So, that is where I think we have some stuff to do. So that it works well, is my perspective.”

P3

P3 elaborated on the ambition with having a process model to apply in projects. On the one hand, P3 mentioned, it can make it easier for consultants to move between projects and know where to find information. Another desired benefit P3 mentioned was for quality assurance and making sure to deliver the right products.
“... to have a structure, to be clear with what we need to develop so that it doesn’t- well, so that we have everything that we should deliver to a customer and that we don’t stand there and don’t have the documentation, the product documentation when it’s all done, because then we have failed too. So, I guess it is a type of quality assurance that we are working in a structured way so that we know what we are doing and deliver the right things.”

P3

P3 expressed the need to adequately maintain and administer the model, as well as keep improving it.

“... it is something we need to keep working on and be clearer about, so we can follow-up better where we are in the process. So, there is a lot to improve, to be even more clear and keep better track.”

P3

P3 also expressed a desire to improve the work with handling requirements, and that there is a need to work with requirements in similar ways across projects. However, still allowing for a certain freedom of choice.

"It is not described in DAD [disciplined agile delivery] as of today, however that is also an activity, that’s written down that we need to describe how we are going to work with requirements elicitation and use cases, and how we should describe requirements, and how should we work according to BDD [behavior driven development] or- so that there is some guideline when the projects get started, that we would like it if you worked in this way.”

P3

“... and then the requirements specification, the work with requirements we have- there we also need to unite on working similarly at least. But we still want to keep this model on such a level that, it is not in detail. There should be a certain freedom within the activities so that- partially so we don’t need to maintain the model too much, but partially so that you can feel that there is a certain degree of freedom to do your work.”

P3

P3 further noted that, at the time of the interview, only a handful of people had much knowledge of the process model and the ambitions for it. The documentation of the process model became known for me as a researcher after the second interview had been conducted. This means that during the first two interviews, no specific questions could be asked P1 and P2 about their knowledge of the documentation and the application of the documented processes. However, P5, who was involved in the two focus projects of the case study in a more remote way, explained that the two projects didn’t exactly apply the defined processes. This was explained to be because smaller projects applied a more light-weight version of the model. P4, however, was asked about the documentation on Disciplined Agile Delivery and Disciplined DevOps. P4 responded that DevOps and Scrum go hand in hand in the projects, although Scrum is more about the daily activities. When asked again about Disciplined Agile Delivery, P4 responded that they were unsure of what it was and added:
4.2. Findings on decision traceability and agility

This section is divided into three sub-sections: practices, attitudes, and challenges. The division into these three categories was a result of inductive reasoning about what general type of theme the coded material fit into. In these categories, different themes that were found to be interesting with regard to the documentation and traceability of decisions are discussed. The practices category includes themes related to the practices used, as described by the informants. The attitudes category includes themes that in the analysis were identified as descriptive of the opinions and attitudes held by informants. The challenges category includes themes that describe aspects described as challenging by informants. The identified themes within each category are the result of inductive analysis. The analysis started with descriptive coding of the interview transcripts after which the coded text passages from all interviews were compared to each other in order to identify similarities and differences. This led to analytical codes being added in order to make the body of data comprehensible. The themes that were then identified and that are presented in this section were deemed to have relevance to the purpose and research questions for the present thesis.

The purpose of this thesis is to understand the importance of decision traceability in relation to product goals and changing requirements in agile software projects.

1. What are the challenges of achieving decision traceability in agile projects?

2. What are important aspects of achieving decision traceability in agile projects?

4.2.1. Practices

Requirements elicitation and traceability

In terms of requirements traceability, no testimony from any one of the six informants indicate that any specifically defined processes or methods are used for assuring traceability from the initial requirements elicitation and through the development of the product. When asked to describe how requirements elicitation was conducted, P1 described a somewhat informal process that set out from some form of discussion with the customer. The discussion with the customer results in user stories, which are then used as a foundation for the tasks that are created in the project backlog in Jira.

“many of these requirements... they become some sort of user story in a way. In the design- they become a part of the design, and what becomes the design becomes some sort of user story, or rather ticket in Jira eventually. And there they are followed up on in the sense that they receive these subtasks that you can check that 'are they done or not' and then they are marked as a pitcher eventually.”

P1

When talking of requirements elicitation, P3 and P4 referred to the behavior-driven development (BDD) model for defining user stories that take the form of statements such as “As a [role], I want to [function], in order to [business value].” The project manager of
the two projects, P4, explained that it differs from project to project how much by the book this type of user story modeling is applied, but that in general, it is the customer who decides what user stories they want. P1 talked a bit about user story models. P1 didn’t specifically mention BDD and noted that they didn’t remember what the user story modeling tool they had used was called. P1 described their experience with user story modeling, as follows:

“….there are some models on how to write user stories when gathering requirements, and we have tried to use some (models), but it didn’t work so- it didn’t go very smoothly, and it was easier just to have a discussion.”

P1

The testimony P1 gives about user story modeling is a bit hesitant, and they express a somewhat skeptical attitude towards its effectiveness. The conclusion P1 seems to draw from their experience of working with user story modeling is that it is easier to just have a discussion with the customer. However, P1 expressed that the user story model worked as a starting point and helped with forming a conceptual understanding from which the discussion could then develop. P1 explained that a probable reason that the user story model hadn’t been applied all the way was that the team didn’t have very much experience with applying the model as a tool and didn’t really know how to use it.

When asked about how initial requirements and documented decisions were followed up on, P1 expressed that they do not usually return to the original text to confirm any alignment. However, P1 did seem to express some confidence in that “it” follows along through the work. This is expressed with the following quote.

“(we) may not return to the original text that was used when it was gathered from the customer at the first meeting or so, but it does follow along in a way throughout the work.”

P1

From this quote, it is not clear what P1 refers to by “it” and what they mean by that it follows along. P1 briefly mentioned that it ends up in the project backlog as a feature, that it eventually is picked up in a sprint, it is worked on, and when it is done it is marked as done. The informant then stressed once again that it is not followed up on in any formal manner.

“any follow-up is not done in the sense that was this- does this still match with what was before, but it is rather so that it follows along forward, I think.”

P1

Documentation and traceability of decisions

One practice applied that relates to decision traceability in the projects is the documentation of important decisions made during the development. Each project had a so-called decision log in its own Confluence workspace. P1 and P5 admitted here as well that it may differ between projects in how decisions are documented, but P1 explained that they try to take notes from meetings with the customer and put in writing what has been decided.
“...we use something called a decision log, in Confluence, where we save all the decisions that get made. And then we also change- if there is something that needs changing in Jira, we change it in Jira, and then we can link them together so that you can see that this Jira has been changed, we have decided on this, and what we have decided, and date, and who were involved in the decision. So it is like a page in Confluence where we save all those things.”

P4

Documenting decisions seemed to be a tool for confirming what has been decided on in discussions with the customer. It was a way for the team to check with the customer that they understood each other.

“I usually try to sort of- try to get some kind of written- either that we have talked through and they get to write back and say that- sort of describe some more what- describe with their own words what we have talked about. So that they get- on the one hand, we get it in writing what they have said and on the other hand we may get more of an understanding that we have understood or not, or if we have understood or not.”

P1

“Usually, I would say, that it is mostly used as... an archive- a receipt that we have understood, and then we look at it in order to make the design. After that, it might not be used very much in the project during the development, but it is used in order to make- iterate the design(...) of the system, and to return to if someone is wondering ‘what have we really said’”

P1

Documenting decisions was often referred to as a type of insurance in case a customer should claim that the product delivered does not reflect what has been agreed upon. During the interviews, this seemed to be important for most of the informants (P1, P2, P3, P5, and P6) as they themselves brought up this aspect of being able to prove who said what. P2, also referred to the legal aspects of fulfilling the contract signed with the customer.

Agile

One topic of the interviews was defining agility and how agile was applied in the projects. P1 compared agile to more traditional projects where many stakeholders are involved and where meetings have to be held in order to make even small decisions.

“... I would say it [agile] is about scaling away unnecessary meetings, to not magnify things that don’t need to be magnified.”

P1

P1 elaborated on the value of this in the following way:

“It’s quicker. There are of course risks to it as well. You may not document- may not realize that we need to document this. But first of all, it’s quicker, and there is more freedom for the developers, that you trust that these people actually can do this in a
competent way. There’s no need to talk with the top managers or get some gigantic machinery going in order to resolve things; we can do it on our own. So, freedom, informality, it’s quick, you work with small pieces at a time, iteratively. There are of course other things formally, but for me, that’s what it means, that’s what I get out of working agile, freedom, informality, and that it’s quick and iterative.”

P1

P2 described agile in terms of working in sprints and the continuous development of a product. P2 also noted that they wouldn’t say that the projects included in the case study were entirely agile.

“an agile project is where you work, first of all, in sprints, and you have a product that you are continuously building, but this is a little fuzzy really, because an agile project, I wouldn’t say that we really are working entirely agile, we work agile-ish.”

P2

Like P2, P6 also mentioned the continuous delivery when describing agile, but added the aspect of delivering value, rather than just a product.

“I think it is about the continuous, what should I call it, the classic, continuous stream of value. That you deliver something continuously, all the time. That it doesn’t go a year between every time you do something.”

P6

P3 stressed that agile is not synonymous with a fast pace and skipping a bunch of things. Instead, P3 argued that working agile might even put a higher demand on iterative development of documentation and traceability issues.

“Agile is not synonymous with working quickly and skipping a bunch of things. It might put even higher demands on that you do smaller iterations of documentation or traceability things, or whatever it may be.”

P3

As evident from the quote from P2 above, P2 didn’t really consider the projects entirely agile. P2 elaborated on this in the following way.

“What we could make more agile since we are doing quite a lot, I mean we are building a complete function in one sprint. I think that if we were to work more agile... we would work more continuously. Instead of working from an order, we would work with the ambition to get this thing done as much as possible, and as quick as possible. We should start with the most critical functions, then work our way towards a finished product, and it can take the time it takes. We do it in the order we can, and we work as fast as we can. Instead, what we do really is that we make a time estimate, this is how the time it is going to take, and it is exactly these things we are going to do, and it is in this order we are going to do it. So, we have quite a lot planned before, or at least a bit more than what maybe scrum would be.”
P2

P2 described that each sprint in the project is a smaller order itself, which gets a time estimate. P2 described the benefits of this as allowing the team, whose members are relatively inexperienced, the opportunity to learn from mistakes when time estimates turn out to be incorrect.

“So, the benefit of working with small scopes is that, if we were to estimate the time incorrectly, we can [realize that] ‘oops, we estimated incorrectly, and it was because of this.’ And now, for us, since we did that the last two estimates... after all, we are rather inexperienced, relatively. But since they are such short sprints, we have the time to learn from our mistakes.”

P2

P2 concluded that for the projects P2 was involved in, the application of agile had more to do with time than anything else.

“agility, I would say that for our part, it is more about time. It is about shorter time spans. But otherwise, I don’t know if it is very agile, really. In fact.”

P2

The testimony from P4 looked somewhat different when it came to the application of agile in the two focus projects of the case study. P4 expressed that the projects were very agile and compared them to another project P4 was involved in.

“in [ProjectX] and [ProjectY], we work very much in an agile way. [CompanyX] uses scrum as a main method... [in] my last project, where I am based at the customer site, it is a bit of a mix of scrum and traditional project management. So, we have our standups, and we have our backlog in Jira, but we also have a management group that we report to with traditional management meetings once a month where we report on what has been done.”

P4

P4 and P5 both commented on the flexibility of agile development when it comes to changes to requirements.

“of course, when you work agile it [changes to requirements] is usually not a big issue because we work in short sprints and we check with the customer all the time that we are headed in the right direction.”

P4

“changing requirements, well, it’s nothing strange. We are working agile. Of course, we should be able to change requirements.”

P5
P6 touched upon the flexibility of agile by mentioning that knowledge is gained during the course of a project that often makes what you thought at the beginning of a project irrelevant. P6 argued that this is the key to why you would want to work in an agile way.

"I guess that is the key to why you want to have agile projects, because, compared to waterfall you have- you think you know what you are going to build, but then when you have completed maybe half of the project, you may realize that what we planned half a year ago is no longer in question. If we were to redo it, we would have built something entirely different, but then you have already gone down a path of how you should build this thing, and you have to keep building it.”

P6

4.2.2. Attitudes

During the coding of the interviews, four attitudes that stood out as important in relation to decision traceability was (1) the importance of clear requirements from the start, (2) responsibility of tracing, (3) focus on what the customer says, and (4) the need for tracing rationale. These four attitudes are presented in this section under their respective subheadings. Quotes from the interviews are presented and described briefly.

The importance of clear requirements from the start

One aspect that was mentioned in the interviews when considering changes to requirements was the importance of conducting pre-studies and making sure to clarify any uncertainties early in the project. P2, P3, and P5 all stressed the importance of resolving uncertainty early and the effect of clarity of requirements from the start.

“... it’s all about preparations really. If we prepare ourselves well, then few questions arise.”

P2

When asked about P2’s experience with changes to the project scope and to product goals, P2 responded in the following way.

“... it is seldom that we or the customer take it so lightly and do such a poor preparatory work so that we might have to do it again... we usually answer most questions before we get started. Partly because we want to do a good job- I mean, if we don’t answer the right questions and we define the goals incorrectly... then we have to pay for it... So, no, I wouldn’t say that- it is usually handled pretty well.”

P2

P3 and P5 also stressed the effect of clarity from the start of a project.

“when the requirements are clear from the beginning, and you know- the customer knows exactly, you know ‘this is what we want,’ then there are seldom any big issues and problems.”

P3
“... for new projects and during the projects I think the preparatory work is essential. A thorough preparatory work usually eliminates most of the problems.”

P5

“Well at least productivity wise, you don’t get as much out of a sprint if you would have done things- prepared things better before they actually end up in the sprint. Because, you might want to, or the customer might feel that ‘we want to get this out quickly,’ so then we squeeze in this one-line requirement. And, well, the consequence of that is that we have to spend the first part of the sprint specifying that requirement, rather than start implementing it right away. So, that is- well, there is really nothing wrong with our process. Because the process itself, or the model, does say that we should have clear requirements going in. That you can’t really put it in the backlog until it is ready.”

P3

Importance of goals

When asked about how to confirm the effect of a requirement’s change on the goals of a project, P1 responded in the following way.

“Well, good question. I think it depends on how you look at it. If we put it like this, if we have discovered that a customer, for example, says that ‘you haven’t made what we wanted,’ then we have somehow by default missed the project goal. And then... what we have written down as the goal of the entire project, is wrong. And then we need to update the goals, maybe, rather than- and then, of course, try to make sure to have a deeper discussion with the customer. That this, what we’re planning now this time actually does match with the goal, and that the goal really matches with what the customer wants.”

P1

P2 highlighted that the most important thing for them as a development team is to know what the customer wants and always to know where they are headed. P2 also stressed that the goal of a project is the most important thing for project success.

”... I would say that what’s absolutely most important- the one hundred percent most important thing in a project, is the goal of the project, that it’s clear. Because if we have a good goal, then we can work well, and then usually most problems are solved. And that you have tried to research the goal, as much as possible. Then 99 percent of projects go well.”

P2

When asked about how to make sure to keep track of the goals, P2 responded in the following way.

“well, on the one hand, it should be documented, but I think that- you need to take the time to be meticulous with it. It is easy to say ‘whatever, we know what this is about,’ but you need to be humble sometimes.”

P2
P6 mentioned the need to revisit the goals of the project continuously in order to readjust them in the case that they should turn out to not be relevant anymore.

“... I think that you all the time, just like revising what should be done next in the development, you should look at, well, are these goals relevant. How do they change- so that you don’t only apply agile in the development team, but all the way up to the chain up to the level of the decision-makers.”

P6

From the quotes above, it may not be clear exactly what P1, P2, and P6 consider to be product goals, but they do seem to acknowledge the existence of goals of a project. P4, however, mentioned that they typically do not work with any product goals or goals on a higher level than the user stories themselves. This may be evidence of different interpretations of what goals are in this respect. Another word that was used by P5 and P6 in a way that seemed to relate to goals was the vision. When speaking of the vision of a project, however, P5 and P6 talked about the challenges of the lack of a vision in a project. This challenge is discussed in section 4.2.3.

Responsibility of tracing

Another aspect that should be important in order to achieve adequate traceability in projects is a clearly defined responsibility of traceability issues and tasks. During the interviews, there seemed to be somewhat of a misalignment between at least two of the informants (P1 and P4) when discussing the need for traceability and keeping track of the impact of requirements changes. The lead developer of one of the projects, P1, responded in the following way when asked about the need for traceability:

“What the requirements are is really not important for me to know. For me as an engineer, not for [CompanyX] as a company, but for me as an engineer, it is pretty unimportant to know when someone said that we would need- that we needed this requirement. What’s important is that there is an updated list- or a list that has the present state of requirements, for me. What existed before is not very interesting for the most part. There may be occasions when it is, but what I can think of now is that- the present is more important than the past.”

P1

This concern for the present state of the requirements, but lesser concern for their origin or earlier versions suggest to me that, from the perspective of a developer, traceability is not such a big concern. This attitude may be contrasted with what the project manager, P4, responded when asked about how to keep track of the impact that changes to requirements may have on the original user stories.

“... that is up to the developers, when discussing the requirements or changes to the requirements, to keep track of really what affects what.”

P4
Focus on what the customer says

Another attitude that surfaced in the interviews was a strong focus on what has been said. This attitude was expressed by P1 as a motivation for documenting decisions and requirements, as evident from the quote below.

“I think it is important to be at least able to go forward and know what has been said. So, decisions, like for example, what functions you want and such things, what requirements the customer has asked for, I think it is important for that to exist.”

P1

The focus on what customers say was also expressed by P2 and P3, but with further stress on the fact that what gets developed is based on what the customer says rather than on anything else. Focus on what the customer says can, in this sense, be interpreted as what the customer is able to express.

“... and we cannot be held accountable for [making] a system to function exactly as the customer wants, we can only make it function in the way the customer has told us it should function... You have to be loyal to what the customer says; you have to count on them knowing what they want.”

P2

“... we do what the customer wants, sort of... We might not understand the [their] organization fully. It is after all the customer who presumably does so. So, I guess we need to rely on what they say.”

P3

P3 noted that the responsibility of the team is the technical aspects of what is to be developed but noted with the quote above that when it comes to other aspects, it is what the customer says that the developers need to rely on. P3 also stressed the documentation of what the customer has asked for in order to achieve traceability.

“Generally, there should always be a customer involved in specifying the requirements, the stories, on some organizational level of the requirements. And it is they who, in that case, initiates the changes, one might think. But it is then all the more important, really, that we keep track of these, and document everything that happens with the changes so that the customer can’t come after the fact and say ‘this is not what I said.’ Well, ‘yes you did, we decided that here, that meeting’ or ‘we wrote this thing.’ So, that kind of traceability is important, to not end up in situations where we might look bad. Now, we shouldn’t blame each other, but it is good to have everything documented so we can keep track of the traceability.”

P3

The need for tracing rationale

Two of the informants (P3 and P6) expressed thoughts that indicate an experienced need for tracing rationale during projects. P3 expressed the following words that point to a need to understand the rationale present when the initial requirements were judged.
“... and then we need to be really sure of how we were thinking when we received the initial requirements.”

P3

P6 expressed a desire to be able to understand requirements and the way in which they have been broken down and implemented in relation to why certain decisions have been made. P6 expressed that the projects lacked this sort of traceability (of rationale).

“... you break down the requirement into a sort of ‘what does this really mean’ and ‘how can we implement this requirement.’ There may be different ways to do that. So to compare that to why we have made certain decisions. That’s where I think the chain is somewhat missing.”

P6

When talking about the need for traceability, P6 once again stressed that why a decision has been made is important to be able to trace.

“well what’s most important, I think, is decisions actually. Because then it is pretty easy to look back and see why did we make this decision, what was the reason, who made the decision.”

P6

The motivation for why P6 finds it important to be able to trace the reason behind decisions is expressed in the following way.

“because I have been in many- well, I have been in situations where a decision has been made, someone, the team makes something, and then you wonder after a while sort of ‘why did we do this’, or someone else comes in and says ‘why are we doing this, we should do it like this instead’. And to then be able to refer to the decision like ‘this is why we did it,’ I think that it’s really good that they have started with these decision logs, I think it’s really important.”

P6

The quotes from P6 above express a desire to trace the rationale behind the interpretation and implementation of requirements. Another point that P6 made is about the rationale behind the requirements themselves. This next quote introduces the idea of considering why the requirements came to be in the first place. In other words, its relation to product goals.

“say we’re talking about- well like we are about doing something with a new payment method now. Then we could look at okay, but what was the reason that we want to implement this payment method, look at what it is with this payment method that we are going to implement.”

P6
4.2.3. Challenges

Five challenges that were mentioned during the interviews stood out as important in relation to traceability within projects. These were (1) customer expectations, (2) assumed consensus with customer, (3) project diversity, (4) fixed price contracts, and (5) lack of shared vision. These five challenges are presented in this section under their respective subheadings. Quotes from the interviews are presented and described briefly.

Customer expectations

One challenge that puts pressure on teams is the expectations that the customer has on the project. P5 stressed that one challenging expectation that customer has is the expectation that pre-studies are redundant. This is for one evident in that customers themselves fail to conduct adequate pre-studies before they turn to CompanyX to develop their software products. P5 expressed this with the following quote.

“when you work with customers, you discover rather quickly that they have, unfortunately, a lack of competence or lack of time to clarify requirements. Actually, Requirements preparation is nonexistent today. These are not specific customers; it goes for all customers. They can’t specify requirements. They can’t conduct real pre-studies before they come to us with an idea.”

P5

P5 further noted that this lack of effort from the customer in specifying requirements leads to that CompanyX becomes responsible for eliciting and specifying requirements and conducting in-depth interviews with the customers. This type of work is something P5 argued they need to get better at within CompanyX. However, P5 also noted that this is a challenge as customers do not want to pay extra for pre-studies.

“no one wants to pay in vain either. Nothing can cost anything. And certainly not pre-studies. Instead, they have to be super quick.”

P5

An even greater challenge when it comes to customer expectations is the expectation that CompanyX won’t need to conduct pre-studies either.

“... they expect that we-no, not that we will need to do it [conduct pre-studies], instead we should immediately understand what they want, and that is the problem.”

P5

P2 and P3 also mention challenging customer expectations, but when it comes to the customers’ inability to foresee the impact that the introduction of new requirements or changes to the requirements have on the project budget. P2 likens it with a person picking candy in a Swedish “pick and mix” candy shop where you pay according to the weight of the candy you pick.

“it becomes almost as if they are walking along the candy shelf- the pick and mix candy shelf, fill up a bag of candy... fine, they pay, and then when they are on their way out, they realize ‘right, I want some candy cars as well,’ but then it’s like hang on, it doesn’t
work like that. You have to take something out so that it weighs the same. So, then you must back up, and you have to remake the requirements.”

P2

P3 noted that this inability to foresee the impact of changing requirements on the project budget means that it is especially important to “keep track of everything.”

“because... there are many examples of that. That we have gotten started, and then we think that the customers themselves realize, when they are a part of the project and say things, that the time spent increases. But they don’t always realize that. So, therefore we have to be clear on that. So, I think then maybe, you notice the most... value from keeping track of everything.”

P3

**Assumed consensus with customer**

Another challenge that has to do with the customer is one that is related to customer expectations, but more so to the communication between the customer and the team. It is the false assumption of consensus between the customer and the team. This may, according to P1, be due to the customer not exactly knowing what they want, resulting in the incorrect definition of product goals. P1 further noted that much of the problems that regard misinterpretation of product goals stem from assumptions made by the team when specifying requirements. Another aspect P1 pointed to is the possibility that the customer might not understand the vocabulary used by the team.

“... so, much of it has to do with assumptions, and the same goes the other way around. That we say that we are going to do x and they [the customer] think it sounds great, it sounds just like what we want, but they might not have understood the vocabulary that we are using.”

P1

P5 also made a comment relating to how the language used can affect the ability to form an understanding between the customer and the team.

“... because misunderstandings occur. We need to find a way where we are talking the customer’s language, and where we find... a place where everyone is happy.”

P5

Another nuance of this type of misunderstanding is when the team assumes that they understand what the customer means by what they are saying.

“So, I think a lot of it is that we maybe assume that we understand each other when we really do not and are not talking about the same thing. I think that’s most common. I do not think it is very often that we don’t bother to listen, I think we usually do listen and we (may write down) what they are saying, and we think that we understand what they are saying, but we don’t.”

P1
Like P1, P3 also mentions that customers often do not exactly know what they want and that the result of this is that initial requirements are often somewhat vague. This may result in assumptions that lead to the development of something other than what the customer wants. P5 expressed the idea that what may often lead to misunderstandings is that the requirements are specified textually.

“well, it is a bit odd because everything is in fact specified. But in continuous text. And that is where the use cases come in actually. Because, with continuous text, things may be misinterpreted. The customer thinks that certain things are implied, maybe, and that is wishful thinking, that things are implied.”

P5

P5 noted that use cases might provide better support, in addition to the textual representation of requirements in order to eliminate the risk of misunderstandings.

**Project diversity**

When it comes to implementing new practices and methods for handling decision traceability one challenge that was mentioned by P3 and P5 was the diversity of projects. All informants mentioned at one point or another during the interviews that the way in which things are handled differs from project to project. P5 mentioned this as a challenge when it comes to process conformity.

“... every project is a new, sort of, situation... There is no key for all- like you can’t work according to a template, so you have to make it up as you go. You have a foundation to stand on, we have a way of working to act according to, but every project has its own prerequisites, depending on the customer, competence, presence, there are so many parameters.”

P5

The aspect of project diversity that P5 mentions above is the difficulty of following specific processes in projects due to their different circumstances. P3 highlights another aspect of project diversity that motivates the ambition to achieve more of a process conformity between projects.

“On the one hand, it may be difficult to jump- if you would need to take in an additional resource from another project. If you work very differently, it may be hard- well, it might not be a huge gap, but it is still harder to come in and understand where to find the information.”

P3

**Fixed-price contracts**

One aspect that came up several times during the interviews as challenging when it comes to handling changing requirements, and to some extent, agility, is the use of fixed-price contracts.
“Customers want to know from the beginning how much it is going to cost, so we always try to estimate from the start about how much we think the work is going to cost, and then that becomes a fixed price for the customer.”

P4

P5 concluded that the use of fixed-price contracts is not optimal, and especially not if pre-studies should become part of the equation.

“... here [the projects] we work with fixed price. The customer orders something. We do a time estimation on it, for a fixed price. And then we deliver it. It isn’t optimal. It’s not. And especially not if we should start doing better pre-studies and produce high quality, and actually make a profit from the deal.”

P5

Furthermore, something that surfaced as possibly challenging with this type of contracting was that it might affect customer involvement. When discussing the resolution of a recent misunderstanding with a customer, P6 mentioned the decision to not have a sprint demo as a missed opportunity for the customer to see what they were paying for. P6 then added the following.

“At the same time, the customer doesn’t really stress over that- because they have a fixed price.”

P6

Lack of shared vision

The possibly greatest challenge when it comes to enabling traceability in the projects has to do with the lack of a shared vision. The customer has the vision for the project, but it’s not explicitly shared with the team. P5 expressed this in the interview the following way.

“there is a vision for [projectX] and [projectY], we are to build something that will become a whole, but right now, we only get small pieces thrown at us.”

P5

“... someone else has the vision for it, the vision is not documented. Instead, it only exists in one head. It is rather fuzzy, and it is not on a detailed level. Instead, it is iterated, and all the time, new information is brought forward. And therefore, we now have changed requirements.”

P5

P6 mentioned the importance of knowing what the team can do to deliver value to the customer. When asked about product goals, P6 expressed the need for having a vision. P6 then mentioned the same issue as expressed by P5 above, and how the customer of [the projects] manages the project tightly.

“..., you need to have some type of vision for it. No matter what you are developing, you need to have some type of vision for what you want to develop and what you want to
achieve. But it’s also about maybe- like in this case it is tightly managed from the customer, like ‘this is what we’re going to do’... it becomes so very focused on, okay, this is what we’re going to do. And there is no room- any leeway for, okay what if we could do it this way instead. So, then maybe we could- so we can get the whole picture.”

P6

P6 mentions the challenge of working towards a goal that is unknown to the team, something P6 mentions as a risk of having vague documentation and unclear requirements.

“... you sit on a bunch of resources that are working towards a goal that you don’t know what it is... that is the risk. That you sort of never- well, you just iterate and iterate, but you never know what you are iterating and where you are going.”

P6

When asked about how to handle the challenge of making the team aware of the goals of a project, P6 responded that team inclusion is important.

“well... I think that the important thing is to include everyone in the team, that the team is aware of- so, it’s not one person who takes the responsibility, but that the entire team is aware of what we have agreed on.”

P6

P6 also stressed the importance of the team being able to look ahead and see what’s coming further down in the project as something that could increase their ability to work in smarter ways.

“if we don’t know what we are going to do, and we don’t know how long it will take, or we don’t have any requirements for further down, we only look at here and now, and we don’t see- don’t look ahead and see that ‘okay, in three months we’re going to do this, okay then maybe we can build this in a smarter way’. Cause I mean, all these people are engineers; they need to know where they are headed in order to- and not just be told, like, ‘build this.’”

P6
5. Discussion

In the first part of the chapter, a note on the term decision traceability for the purpose of the present thesis and how it is handled in this chapter is given. In the second part of the chapter, a discussion of empirical findings in relation to the previous research presented in the literature review is given. The discussion sets out from the two research questions which are discussed separately.

5.1. A note on decision traceability

The purpose of this thesis was to understand the importance of decision traceability in relation to product goals and changing requirements in agile software projects. For this purpose, two research questions were developed. The first one is aimed at identifying challenges of achieving decision traceability in agile projects, and the second is aimed at identifying important aspects of achieving decision traceability in agile projects. Before moving on to discuss each question individually, I would like to make a claim about decision traceability and how I will approach the subject in this chapter. I would like to start with looking back at what I said about decision traceability in the first chapter and the problem definition of this thesis.

“For the purpose of this thesis, I will, [...] use the term decision traceability to refer to the ability to trace decisions that relate to the evolution of a software product, as well as the fulfillment of product goals. This will include traceability of requirements, but also the decisions that the specified requirements are a result of, as well as design decisions.”

From problem definition, section 1.1.1

As mentioned in the literature review, Gotel and Finkelstein (1994) explain that pre-RS traceability allows for decisions and closed issues from the early requirements specification activities to be re-opened and re-worked. In other words, pre-RS traceability makes decisions from the requirements elicitation and specification traceable. With the above description used for decision traceability in the problem definition and Gotel and Finkelstein's (1994) definition of pre-RS traceability, it is clear to me that pre-RS traceability is an important part of decision traceability. I make this claim since the relation between product goals and specified requirements can only be made traceable through pre-RS traceability. Without pre-RS traceability, the motivations behind the existence of requirements will have to be the result of reconstructions, just as design rationale would have to be reconstructed in order to understand the design if there is no design rationale traceability (Tang et al., 2007). Since pre-RS traceability only covers the early requirements specification activities, I will also argue that traceability of rationale behind requirements and decisions made during the rest of the project, is also an important part of decision traceability as I refer to it in this thesis. With this clarification of my take on decision traceability, I hope to provide traceability as to my reasoning in answering the research questions below.

5.2. Answering the research questions

Drawing from the findings presented in the previous chapter, as well as the literature review in chapter 3, I will now turn to answer the research questions. An overview of the different topics discussed in answer to the research questions is given in Figure 7. The left column shows the challenges identified in the answer to the first research question. The small arrows between the challenges illustrate their potential interconnectedness, which I will also discuss. The right
column in Figure 7 shows the important aspects of achieving decision traceability as identified in answer to the second research question.

![Diagram showing challenges and important aspects of achieving decision traceability]

Figure 7: Challenges and important aspects of achieving decision traceability in agile projects

5.2.1. What are the challenges of achieving decision traceability in agile projects?

As a result of the analysis of empirical findings, I have identified seven challenges that stood out as important obstacles to overcome in order to achieve decision traceability in agile projects. I discuss each of these challenges separately below.

**No habit of ensuring pre-RS traceability**

The first challenge to discuss has to do with habit and experience. As mentioned in the empirical results, no testimony from any one of the informants suggested that there is a habit of ensuring traceability from early to later requirement specifications. P1 explicitly stated that any follow-up on the original text gathered from the initial discussions with the customer is not usually done. This is a statement that to me speaks clearly about the lack of habit in ensuring traceability from early requirements to the then specified requirements or user stories. This may be a serious challenge. Especially considering that inadequate or lacking pre-RS traceability has been recorded as the cause of most problems assigned to poor requirements traceability (Gotel & Finkelstein, 1994).

P1 spoke vaguely about some aspect of the early formulations of requirements following along throughout the work, but no specific examples of how was given. From my point of view this may mean one of two things: (1) there is, in fact, a way that early requirements
follow along in the ongoing project work, only P1 wasn’t able to explain how during the interview, or (2) the statement made by P1 merely reflect a faith in that the requirement transformation process, by itself, ensures that early requirements are made justice and are reflected by the resulting user stories and tasks. Either one of these two explanations may be true; in fact, both might be true. My point is, however, that no matter which one of these interpretations you choose to believe, it indicates to me a challenge in achieving pre-RS traceability, and thereby decision traceability. Let me explain my reasoning.

Interpretation 1 – There is, in fact, a way that early requirements follow along in the ongoing project work, only P1 wasn’t able to explain how during the interview. In the case that this interpretation is correct, I see at least three possible explanations (there may be more) for why P1 couldn’t explain how early requirements follow along in the project work. The first explanation I can think of is (a) P1 was unsure of exactly how the early requirements are evident in the requirement transformation and the ongoing project work. This would be a problem since P1 is a developer and even the lead developer of a project. If the lead developer is unsure of how early requirements are reflected in the later specified requirements and the ongoing work of the project (pre-RS traceability), then there is a lack of pre-RS traceability in the project. The second explanation that comes to my mind is (b) P1 simply meant that the early documentation or meeting notes from the initial customer discussions, referred to as ‘original text’ by P1, are available to the team in the digital project workspace throughout the project. This may be a good start for achieving pre-RS traceability, but as Gotel and Finkelstein (1994) note, the documentation of required information is no guarantee of its traceability. And even if it is structured in a way that makes it traceable in many ways, there is no guarantee that it will be up to date (Gotel & Finkelstein, 1994). Therefore, simply storing meeting notes from early discussions with the customer does not guarantee decision traceability. It doesn’t ensure that the ideas represented by the early requirements are considered or remembered when the resulting user stories are adapted or changed later in the project. Furthermore, P1’s statement that there is no follow-up on the original text indicates that at least P1 does not typically take on the responsibility of checking that original intentions for the project are met by the ongoing requirements negotiation. This takes us to the third explanation, (c) P1 was unsure of how the early requirements follow along in the ongoing project work as it is not P1’s responsibility to know of or ensure pre-RS traceability in the project. I will not elaborate any further on this interpretation here, as the responsibility of tracing was identified as a challenge in itself. This challenge is discussed in the next subsection.

Interpretation 2 - The statement made by P1 merely reflect faith in that the requirement transformation process, by itself, ensures that early requirements are made justice and are reflected by the resulting user stories and tasks. In the case that this interpretation is correct, I believe the problem to be rather clear. Simply having faith in that early requirements are sufficiently reflected in the transformation to user stories, and the further project work is no guarantee that so is the case. It is wishful thinking at best. At worst, it may be the cause of major project failure. If interpretation 2 is correct, then it may reflect a symptom of the value perception issue of pre-RS traceability as identified by Williams (2014). If the value of achieving pre-RS traceability is not perceived by developers, then they won’t engage in the required activities to achieve it. And if the lead developer of a project has faith in that the pre-RS traceability will take care of itself, as a result of the requirement transformation process, then the value of putting effort into ensuring pre-RS traceability is evidently not perceived.
Unclear division of responsibility

As mentioned earlier, the empirical findings suggest that the responsibility of traceability is a challenge. This is evident in the empirical findings, no matter the interpretation one made of the previously discussed statement made by P1. As mentioned in the previous section, the lack of an explanation for how P1 believes early requirements follow along with the ongoing work of the project can be explained by P1 not having the responsibility to know how pre-RS traceability is ensured in the project. But even if this is not the explanation you choose to go with, another statement made by P1 suggests that P1 does not quite take responsibility for ensuring traceability of changing requirements. Let’s return to a quote that was presented in section 4.2.2 under Responsibility of tracing.

“What the requirements are is really not important for me to know. For me as an engineer, not for [the organization] as a company, but for me as an engineer, it is pretty unimportant to know when someone said that we would need- that we needed this requirement. What’s important is that there is an updated list- or a list that has the present state of requirements, for me. What existed before is not very interesting for the most part. There may be occasions when it is, but what I can think of now is that- the present is more important than the past.”

P1

I briefly commented on my interpretation of this quote in the previous chapter, but I now want to elaborate on what I said earlier. When this quote was presented in the previous chapter, I noted that it suggests that from the developer perspective, traceability is not such a big concern. Well, I would now want to take the interpretation a step further and argue that this statement from P1 may suggest that P1 does not consider it their responsibility to ensure traceability of changing requirements. P1 does express that it is important that there is an updated list of requirements, but that what existed before is, for the most part, not interesting. At least not for an engineer.

The “for me as an engineer” part of the statement suggests that P1 recognizes that early requirements may be important for someone, but not for someone with P1’s role. P1 does not mention who it may be important to other than the organization itself. But the organization itself is not able to take responsibility and perform tasks ensuring decision traceability. Therefore, If the lead developer does not assume this responsibility, who should? Well, just as I mentioned in the previous chapter, this may be contrasted to the statement made by P4, who was the project manager for the project in which P1 was the lead developer. If the lead developer does not assume the responsibility of ensuring traceability, one might think that the project manager would. But the project manager explained that it is up to the developers to keep track of what affects what when discussing changes to requirements. Keeping track of what affects what arguably sounds like a traceability task. A task that the project manager ascribes to the developers.

The question is then, of course, is there really a clear division of responsibility regarding traceability? As evident from my discussion above, my interpretation is that there is not. The lack of allocation and management of different roles that practitioners need to assume – in order to obtain, document, organize, and maintain required information – is another problem mentioned by Gotel and Finkelstein (1994) as a cause of poor pre-RS traceability. The empirical findings from the present case study do indeed indicate that there is no clear allocation of roles regarding traceability in the projects. The project manager maintains that it is the developers’ role to keep track of things, while one of the
two lead developers only seem to be aware of their role as an engineer, to whom only the current state of requirements is interesting. This attitude may once again be an indication of poor value perception of traceability, as discussed by Williams (2014). I will return to discussing the issue of value perception in section 5.2.2.

P1 recognized that there might be occasions when earlier requirements are interesting but couldn’t give an example of such a situation in the interview. This may indicate what Gotel and Finkelstein (1994) mentioned as yet another problem faced by providers in achieving pre-RS traceability, namely that individual efforts in achieving pre-RS traceability are ad hoc and localized. Gotel and Finkelstein (1994) highlight that this is a problem because combined and full-time responsibility by all is really needed.

**Lack of rationale documentation**

As presented in the empirical findings, P6 expressed a lack of a chain connecting a requirement to the interpretation of its meaning, how it should be implemented, and why certain decisions have been made. Noteworthy is that at the time of the interview, P6 had only been involved in the projects for a few weeks. Now, the significance of this you may interpret however you like. Either P6’s late introduction to – and short experience of working in – the projects meant that P6 was not familiar enough with the projects at the time of the interview to give an accurate testimony on their traceability. On the one hand, P6 might not have been able to account for the processes and activities used in the earlier phases of the projects – which P6 was not present for. On the other hand, you may look at P6’s testimony in this case as rather interesting as P6 came to the projects with a set of fresh eyes. This should have made P6 particularly equipped to judge the ability to follow the rationale behind requirements changes, and design decisions, in the documentation produced by the projects. The fact that P6 is the one who expressed a lack of traceability of rationale should not be surprising as justifications of decisions often exist implicitly in the heads of those who made them (Liang et al., 2010; Tang et al., 2007; van der Ven et al., 2006). Since P6 was new in the projects at the time of the interview, decisions had been made prior to P6’s introduction to the projects. Important decisions had been recorded in the decision log of each project. However, P6, who for obvious reasons did not have access to the justifications of these decisions implicitly, experienced a lack of traceability of rationale in the decision logs.

That P1, P2, and P4 did not mention a lack of rationale documentation or traceability, may very well be that they felt they understood the rationale behind any decisions that had been made in the projects. This could simply be explained by the fact that they had been involved in the decisions. The suspicion that the rationale behind decisions made during the projects existed implicitly in the heads of team members is supported by the fact that most of the informants seemed to consider the decision logs to document primarily what had been decided. P1 described it as a receipt that they would return to if someone wondered what had been said. The lack of a mention of the why behind the decisions, from any of the other informants who were directly involved in the projects (P1, P2, and P4), is also interesting. It suggests that documenting the reasons for decisions made during the projects is not a primary concern. The focus on what over the why is something that I consider a challenge in need of its own discussion. I, therefore, elaborate on this challenge in the next subsection.

The empirical findings show that the lack of documentation of rationale has proved to be a challenge for a newly introduced team member and project manager. If we now turn to
what previous research has recorded, the lack of traceability of design rationale may cause several other problems as well, some of which are especially important in achieving decision traceability in relation to requirements changes and product goals. As mentioned by Tang et al. (2007), business goals and constraints might be ignored; it might be unclear how design criteria and environmental factors influence the architecture; design integrity might be violated, and inaccurate assessments may be made of the impact of changing requirements and environmental factors. Considering the empirical findings and the theoretical arguments for documenting decision and design rationale, the lack of rationale documentation is indeed a challenge that needs to be overcome in order to achieve decision traceability.

**Tendency to focus on the what over the why**

As I mentioned earlier, there seemed to be a stronger focus on the what of decisions, than on the why behind them. This is evident from the empirical findings describing the documentation of decisions as well as those indicating a focus on what the customer says. Now, the findings presented on the focus on what the customer says, might not have been very interesting or surprising at all had it not been for the lack of the same amount of stress by informants on understanding the why behind what the customer says. The fact that the documentation was only ever spoken of as a means of keeping an archive of what had been said and agreed upon, as a result of discussions with the customer, struck me as surprising. Especially considering that many decisions and trade-offs need to be made in order to realize the requirements (Cleland-huang et al., 2013; Mirakhorli, 2011; Tang et al., 2007; van der Ven et al., 2006), and the problems that may arise when the rationale behind decisions are not made traceable (Tang et al., 2007).

The strong focus on the what over the why is expressed by P2 by stating that the team cannot be accountable for making a system function exactly as the customer wants, only the way the customer has told them it should function. P3 expresses the focus on the what by explaining that since the customer should know their own organization better, the team must rely on what the customer says. These findings are especially interesting when contrasted against a challenge that P5 brought up. Namely the fact that all customers are unable to conduct adequate pre-studies and specify requirements. This, together with the findings that suggest that assumed consensus with the customer, and resulting misunderstandings, are quite common, indicates to me a potentially rather devastating problem with focusing primarily on what customers say rather than the why's behind.

Furthermore, it occurs to me that the focus on the what over the why does not fit very well with some of the recorded attitudes expressed by informants. I refer to the stressed importance of clear requirements from the start and the importance of goals. P2, P3, and P5 expressed the importance of resolving any uncertainty early, and the effect of having clear requirements from the start. P2 even stated that they usually answer most questions before they get started and that they seldom do such a poor preparatory work that they might have to do it again. But at the same time, misunderstandings do occur as a result of assumed consensus on the meaning of the things that have been said. The importance of goals, as expressed by P2 as the most important thing in a project, also seems to contradict the focus on the what over the why. My intention when I bring this up is not to undermine P2’s intellectual understanding that goals are important for project success. However, the understanding that clear goals are important might not be reflected in the everyday work of the teams. I agree that clear goals are important in order to achieve decision traceability, but I will also argue that focusing on the what over the why is not consistent with
achieving decision traceability as business goals and constraints might be ignored as a result of rationale behind decisions not being adequately documented and traceable (Tang et al., 2007). Furthermore, focusing on the resulting artifacts from the architecture design process rather than the decisions that lead to them may cause architectural drift and software erosion (van der Ven et al., 2006). The why and the what will be further discussed in section 5.2.2.

**Unrealistic customer expectations**

The existence of unrealistic customer expectations was something that P5 stressed in the interview. The challenge that P5 explained was that customers are, on the one hand not able to sufficiently prepare requirements themselves, and on the other hand, they do not expect the consultancy firm they turn to for developing their product to need to do any extensive pre-studies. It is challenging because they expect the team assigned to the project to be able to immediately understand what they want and need, and that is the problem, as stated by P5.

P2 and P3 highlight another unrealistic expectation that customers tend to have. This was the inability to understand or foresee the impact that added or changed requirements may have on the scope or budget of the project. P2 likened it to picking candy in the candy shop and then wanting to pick even more after paying for the weight of the already picked candy. P3 mentioned that customers don’t always realize that as they introduce changes or new requirements, the time spent on the project increases. In other words, the tradeoff between budget and scope is not always considered by customers when changes are introduced. This is a challenge that Rasmusson (2010) recognizes as he stresses the importance of setting the right expectations at the outset of a project regarding what has to give. This is further discussed in section 5.2.2.

This challenge is related to a project aspect that was mentioned as challenging by informants, the fixed price contracts. The fixed price contract was something that, at least partially if not entirely, seemed to be applied as a result of customers wanting to know beforehand how much a project is going to cost. This expectation from the customer was expressed by P4. P5 expressed that the fixed price contract is not optimal, and certainly not if conducting more comprehensive pre-studies is an ambition. Thus, the expectation of fixed-price contracts may also be an issue standing in the way of achieving decision traceability. According to Rasmusson (2010), the scope is best kept flexible in agile projects as budget, time, and quality tend to be more fixed aspects of a project. The fact that the projects apply fixed price contracts should thus mean that the scope should be kept flexible. Yet it seems this relationship between budget and scope may be misunderstood by customers.

**Lack of shared vision**

The previous discussions on documentation of rationale, focus on the what over the why, and the inconsistency pointed out in relation to the importance of goals, makes yet another challenge relevant to bring into the conversation. This is the lack of a shared vision in the projects. As presented in the empirical findings, P5 described that the vision for the two projects in the case study only exists in the head of the customer. The vision is, as described by P5, rather fuzzy and bit by bit, it is revealed to the team. P5 also noted that this had been the cause of changing requirements during the projects. The problem described by P5 does not come as a surprise when reviewing the literature. As explained by Jayatilleke and Lai (2018), changing requirements are often triggered by the project...
vision becoming clearer, or that the project space becomes better understood from a customer point of view.

This is a challenge to achieving decision traceability as, without access to product goals or “the vision” behind the project, it becomes hard to both ensure pre-RS traceability (Gotel & Finkelstein, 1994) and understand the rationale behind decisions (Tang et al., 2007). This leads me to think there is another point to be made here. Considering that the project vision hadn’t been sufficiently communicated and shared with the project team, it might not be very surprising that the empirical findings suggest a strong focus on what the customer asks for, rather than the why behind. Now, I do not have the evidence to support what I am about to suggest. After all, I do not know how the initial discussions with the customer went, but let’s speculate for the sake of my argument. Could it be that the reason for the strong focus shown by informants on the what over the why of requirements and decisions, is an effect of the customer not sharing the whole vision for the project upfront? And the result is poor pre-RS traceability and lacking documentation of rationale. Of course, there is the possibility too that the customer hasn’t shared the vision for the project, because they simply haven’t been asked and they didn’t see the need to, as a result of the teams not putting much effort into understanding the why. Or it may simply be possible that – as Jayatilleke and Lai (2018) explain – the project vision or the problem space became clearer for the customer as the project developed. This was something that P6 mentioned as a motivation for doing agile projects. Because the plan you start out with becomes irrelevant as more knowledge is developed during the project.

Which interpretation is most likely is, however, not my point. I simply want to point out the potential interconnectedness of the different challenges I have identified. This strikes me as important as the challenges of achieving decision traceability are most likely not challenges that can be eliminated one by one. In some way or another, they are very likely to relate to each other. Cause and effect may or may not be important to examine, but being aware of the complexity and interconnectedness of the different challenges should be important in order to tackle them successfully. What the empirical findings do suggest, is that there is a tendency in the teams to focus more on the what than the why behind requirements and decisions; that the rationale behind decisions and requirements are not always sufficiently documented and traceable; and that there is a lack of a shared vision in the projects. Whichever came first is hard to say from the empirical findings, but in my mind, they are very likely dependent on each other.

**Assumed consensus and misunderstandings**

Assumed consensus with the customer and resulting misunderstandings was mentioned as a challenge by the informants themselves in the interviews. Assumptions being made by both the customer and the team was mentioned by P1. P1 and P5 both mentioned the aspect of vocabulary and speaking the customer’s language as a challenge. The assumption of consensus where none exists is, according to Rasmusson (2010), what kills most projects.

The challenge of assumed consensus may in part be a consequence of customer expectations, as discussed earlier. The customers assume that the team will immediately understand what they want, which in turn may cause misunderstandings. In discussing the importance of goals, P2 mentioned that it is important to be humble and not assume right away that you know what the customer is asking for. In considering the challenge of assumed consensus, we now find ourselves in another discussion of the
interconnectedness of the different challenges in achieving decision traceability. As highlighted by Book et al. (2016), transparency and communication are two of the key principles of every project. The lack of a shared vision may result in assumed consensus and misunderstandings, but the assumption of consensus might just as well lead to the vision not being explicitly communicated. The assumption of consensus may also lead to the perception that the rationale behind requirements and decisions don’t need to be documented and made traceable. And as the rationale behind decisions and requirements don’t get documented, it may be harder to detect potential misunderstandings and so instead they are enforced by the assumption of consensus.

These challenges are highly interconnected, as I have mentioned earlier. There may be more connections to be made and even more challenges to consider. But the challenges I have mentioned and discussed above are the ones that the empirical case study of the present thesis has me convinced to be important ones to tackle in order to achieve decision traceability in agile projects. After identifying these seven challenges, I asked myself how they can be further categorized to get a grasp of how they may be tackled in practice. When it comes to tackling these challenges, I consider them to be mainly of two distinct types; practice-related, and communication-related. By practice-related challenges I refer to challenges that may be handled by improving or implementing practices that better support decision traceability, and by working on team engagement and involvement. By communication-related challenges I refer to challenges that require improved or new ways of communicating with the customer. The division of challenges into practice-related and communication-related challenges is illustrated by Figure 8.

![Figure 8: Practice-related and communication-related challenges of achieving decision traceability](image)

5.2.2. What are important aspects of achieving decision traceability in agile projects?

As you may have noticed from the discussion on the first research question, the identified challenges do highlight important aspects in achieving decision traceability. Answering the first research question is, therefore, part of answering the second one. This will soon be evident as many of the important aspects in achieving decision traceability that I will now turn to are based on discussions that I started in the previous section. In answering this research question, however, a broader and more solutions-oriented discussion is in order.

**Responsibility and team involvement**

In answering the first research question, the challenge of an unclear division of responsibility regarding traceability was discussed. As I mentioned then, Gotel and Finkelstein (1994) stress that a combined and full-time responsibility is needed in order
to achieve pre-RS traceability. I believe the same to be true when it comes to documenting the rationale behind design decisions as well. Combined and full-time responsibility is one aspect. However, concluding that everyone on the team has the same responsibility to ensure decision traceability seems unlikely to be enough. First of all, there already seems to be a misalignment of one of the lead developers and the project manager. These perceptions of responsibility might not disappear because it is simply agreed that everyone holds a responsibility. Secondly, Gotel and Finkelstein (1994) highlight that the different roles that practitioners need to assume in order to ensure pre-RS traceability, need to be managed and allocated. In other words, there should be no room for team members to sign traceability tasks off by assuming that they are someone else’s problem. P6 mentioned team involvement when discussing the importance of project goals. Involving the whole team when setting project goals also means creating opportunities for improving the pre-RS traceability and traceability of rationale. Including everyone involved in the early discussions of important decisions is also an important part of project success (Book et al., 2016; Rasmusson, 2010).

Recognizing that involving the whole team in ensuring decision traceability is important takes us back to an aspect I earlier promised that I would come back to; value perception. If developers do not see the value of putting the effort necessary into ensuring pre-RS traceability, they will not do it (Williams, 2014). An important part of involving the whole team in taking responsibility for ensuring decision traceability is thus addressing the value perception issue. Clarifying the intentions behind improving the pre-RS traceability, as well as the traceability of rationale behind requirements and design decisions, is one way of addressing the value perception issue. One argument that may help sway both developers and managers into seeing the benefit of traceability is the findings suggesting that traceability increases developers’ performance (25% in terms of speed and 50% in terms of correctness) (Mäder & Egyed, 2015) as well as software quality (Rempel & Mäder, 2017).

**The why before the what**

In answering the first research question, the focus on the what of requirements and decisions over the why behind them, was discussed quite a bit. Therefore, it should come as no surprise to the reader that I will now argue that putting the why before the what is an important aspect in achieving decision traceability in agile projects. The vision for the project was mentioned by P6 as important for the team to be able to look ahead on what’s to come further down the line in the project. This was stressed as important by P6 in order to allow the team to make smarter decisions when developing the software product. The importance of communicating the goals, vision, and context of the project is stressed by Rasmusson (2010), who also highlights that it will help the team make smarter decisions. The vision, goals, and context are the whys behind a project. Now, arguably, looking ahead and being able to make smarter decisions does neither guarantee nor depend on decision traceability. However, knowing the why behind a project or a product that the project is to deliver, makes it possible to document the rationale behind requirements and decisions. When the rationale gets documented, opportunities for ensuring decision traceability arise. After all, traceability is the access we create for ourselves to the essence of a product (Palmer, 2014). Therefore, ensuring decision traceability, by improving both pre-RS and rationale traceability, creates access to the essence of the product, allowing the team to form and continually check and revise their “big-picture” understanding of
the project. The “big-picture” understanding that starts with asking why are we here? (Rasmussen, 2010).

Putting the why before the what means responding to two previously discussed (section 5.2.1) challenges, in particular. The first one is quite obvious and shouldn’t require much explanation; by putting the why before the what we respond to the tendency to focus more on the what than the why. Now, this may be easier said than done since it will most likely require a shift of mindsets rather than practices. To use words of Kulak and Li (2017), it is about the right view and right intention rather than the right speech and right action. In terms of being agile, putting the why first would mean putting a bit more emphasis on the value-driven element of agile (Moran, 2015).

The second challenge that putting the why before the what responds to is the lack of rationale documentation and traceability. As mentioned earlier, focusing more on the what creates opportunities to document the rationale behind requirements and decisions. If we are able to shift our mindsets to the right view and right intention by becoming more value-driven, the likelihood that the rationale behind requirements and decisions get documented increases. Because the team then recognizes the importance of making rationale traceable. By bringing the right view and right intention into the discussion I mean to highlight that achieving decision traceability will first require the team to want to achieve it because it has value, not just because a certain process says so. An example of what the right view might look like is that decision traceability is something we need, for it helps us understand the customer and the product they seek to develop. An example of what the right intention when pursuing decision traceability might look like is that we do it because we want to understand the customer so we can build the right product for them. Now, I realize that this last point may be harder said than done. We find ourselves back at the value perception problem (Williams, 2014). Team involvement, as discussed earlier, is thus an aspect that needs to be addressed together with this one.

I have now addressed how focusing on the why before the what is a way of addressing two of the identified challenges directly. However, these are not the only challenges that this shift of focus will help address. It will also help minimize the risk of two other challenges identified in the case study. The first one is the lack of a shared vision. By showing an increased interest in the why behind a project, the vision held by the customer may be unveiled and shared with the team more quickly. The other challenge that the shift from the what to the why will help prevent is the challenge of assumed consensus and resulting misunderstandings. By engaging in discussing the why behind requirements and decisions with the customer, potential misconceptions can be resolved, and devastating misunderstandings can be avoided. These last two points take us to the next important aspect of achieving decision traceability; fostering healthy expectations for decision traceability.

**Fostering healthy (customer) expectations for decision traceability**

One of the challenges discussed in answering the first research question was the unrealistic expectations that customers tend to have. The expectations that were mentioned were regarding the effort and time – and therefore also budget – needed to specify accurate and sufficiently clear requirements. The tradeoff between scope and budget when changes are introduced was also an issue where the expectations of customers were experienced as somewhat unrealistic, or simply uninformed. By setting healthy expectations, I mean setting expectations that will allow the team and the
customer to achieve what they want to achieve together. First of all, the expectations from the customer need to be compatible with the expectations held by the team. Secondly, if the ambition is to achieve decision traceability, expectations need to be set that allow room for whatever activities that may be needed to achieve decision traceability. Healthy expectations in this sense means expectations that create opportunities to achieve what we want to achieve, rather than creating obstacles.

This challenge calls for attention to setting expectations with the customer that are compatible with those of the team. At the outset of a project, this means sitting down with the customer and having a discussion on the tradeoff between budget, time, scope, and quality. This is an important expectation-setting activity advocated by Rasmusson (2010) and Goldkuhl and Röstlinger (2017). Rasmusson (2010) further notes that scope is the aspect best suited to being kept flexible as budget, time, and quality tends to be more fixed. As fixed-price contracts were applied in the projects of the case study, and informants mentioned the challenges it involved, setting the right expectations with the customer from the beginning of the project, stressing that the scope may have to decrease, should help get the projects on a better path for success.

The fact that customers tend to underestimate the time and effort required for adequate requirements elicitation and specification indicates to me that this is another thing that needs to be communicated about with the customer before the project gets started. This also relates to the previous discussion of putting the why before the what. If the ambition of the team is to become more value-driven and put more effort into ensuring decision traceability, then the customer needs to be in on what’s expected of them in order for the project to achieve this. By sharing the intention to create a higher quality product for the customer by forming a “big-picture” understanding of the problem space and the product goals, I think the customer will be more likely to share the product goals and the vision behind the project. By sharing the intentions, the team can help set customer expectations that; eliminate the challenge of the project vision and goals not being shared – thereby minimizing the risk of misunderstandings; and increase the team’s ability to ensure decision traceability.

In my opinion, a potentially great way of getting started with aligning the team’s and the customer’s expectations is the inception deck described by Rasmusson (2010). The activity itself can be an expectation-setting tool by showing everyone involved – the customer as well as the team – what is important to know about the project before it gets started. The different parts of the inception deck will, among other things, help define the why of the project, the scope – what the product is and what it isn’t, what it’s going to take to build it, as well as priorities between aspects such as budget, time, scope, and quality.

**Project particulars**

The inception deck is a way of getting to know the particulars of a project. There may be other factors than those included in the inception deck that may affect the ability to achieve decision traceability. The fixed price contract is one such aspect of a project, which was mentioned in the interviews as a potentially challenging aspect of handling changing requirements. The diversity between projects was also brought up by informants as something making standardized processes difficult to apply and motivate. I, therefore, want to highlight project particulars as an important aspect of achieving decision traceability. While considering the previously mentioned aspects, the team members also
need to consider how the particulars of the project will affect the ability to achieve decision traceability. It may have to do with resources, availability of customer, or ability to obtain information (Gotel & Finkelstein, 1994).

**Clarifying an agenda for requirements elicitation and pre-RS traceability**

As discussed in answer to the first research question, there seemed to be little or no habit of ensuring pre-RS traceability in the projects. A need was also expressed by P3 to unite on working similarly across projects with requirements specification. P6 and P3 both mentioned being able to move between projects and know where to find the right information as a benefit of working according to the same processes across projects. Clarifying an agenda for what the requirements elicitation process should look like, what format requirements should take, and how to make the rationale behind requirements traceable, would help the team get into the habit of ensuring pre-RS traceability.

My suggestion is that the work on defining processes for requirements elicitation and pre-RS traceability should start with considering the different aspects of achieving decision traceability, which I have discussed above. In particular, this should also involve considering how the defined processes or models can make room for flexibility and adaptation to project particulars. Like P3 and P5 mentioned, there is no one size fits all, and the process model defined should allow for certain flexibility in the projects. This flexibility is important for achieving agility as empowering teams to be self-organized is a core element of agile (Moran, 2015).

The aspect of clarifying an agenda for requirements elicitation and pre-RS traceability takes us to the last aspect that I will discuss. Namely, ownership of process improvement activities and documentation.

**Ownership of process improvement activities and documentation**

Having the ambition to define and improve project processes is good if you want to achieve decision traceability. But in order to realize those ambitions, someone needs to initiate the changes and communicate them to everyone it may concern in the organization. P3 mentioned that the ownership of the process model documentation is not clearly defined and that someone needs to be appointed to carry the work with it forward. This ownership should need to be clarified before any agenda for requirements elicitation and pre-RS traceability can be realized. It also seems important to figure out how the process model can be successfully shared and communicated to everyone in the organization. At the time of the case study, few people knew of the process model documentation. P3 explained that this was due to the lack of clear ownership regarding the documentation and activities around the development and administration of the process model. Appointing someone to take ownership of the development, administration, and communication of the process model should thus be important. Especially since poor feedback regarding best practice, and little dedicated procedural support has been recorded a problem restricting advances in pre-RS traceability (Gotel & Finkelstein, 1994).

Just as the challenges of achieving decision traceability may be divided into two types of challenges, I propose a division of the important aspects identified into three types of aspects. These different types have to do with in what context these aspects may be dealt with. The first type of aspect is that of team engagement. Team engagement aspects are the aspects having to do with responsibility and team involvement, and focusing on the *why* before the *what*. These
aspects have more to do with forming and engaging the team in order to enable decision traceability. The second type of aspect is project-specific aspects. These aspects are aspects that need to be revisited and kept in mind with every new project. They are aspects that may vary between projects, and thus, how decision traceability will be achieved with those specific aspects in mind will need to be considered. The project-specific aspects are those project particulars that may affect how decision traceability may be achieved, as well as customer expectations. The last type of aspect is process improvement aspects. They have to do with achieving decision traceability within a longer time frame. They are clarifying an agenda for requirements elicitation and traceability, and ownership of process improvement activities. The division of aspects into these types is illustrated in Figure 9.

Figure 9: Team engagement, project-specific, and process improvement aspects of achieving decision traceability

5.3. Summary of discussion

In this chapter, I have presented a discussion on my take on decision traceability in relation to changing requirements and project goals. As a result of the literature review, Pre-RS traceability and the traceability of rationale behind requirements and decisions were identified as important parts of decision traceability. From this definition, the discussion then took off from the two research questions developed for the present thesis. In the discussion of the findings, seven challenges of achieving decision traceability were argued to be found. These challenges were argued to be either practice-related or communication-related. Six important aspects of achieving decision traceability were discussed in answer to the second research question. These aspects were regarded to be either team engagement aspects, project-specific aspects, or process improvement aspects. The interconnectedness of the various challenges and aspects were also stressed. I argued that overcoming the challenges identified would require considering how they may affect each other and not just managing them one after the other.
6. Conclusion

In this chapter, conclusions about the findings, implications, and knowledge contribution of the present thesis are presented.

6.1. Contributions

The problem that I set out to investigate in writing this thesis was the problem of achieving decision traceability in agile projects. The term decision traceability was used in this thesis to refer to the ability to trace decisions that relate to the evolution of a software product, as well as the fulfillment of product goals. This included both pre-RS traceability (the ability to trace specified requirements back to their origin) and rationale traceability (the ability to trace decisions made in terms of the reasoning behind them).

The purpose of the thesis was:

*to understand the importance of decision traceability in relation to product goals and changing requirements in agile software projects.*

The research questions the thesis was designed to answer are:

1. What are the challenges of achieving decision traceability in agile projects?
2. What are important aspects of achieving decision traceability in agile projects?

6.1.1. Challenges of achieving decision traceability in agile projects

In answer to the first research question, seven challenges were identified:

- No habit of ensuring pre-RS traceability
- Unclear division of responsibility
- Lack of rationale documentation
- The tendency to focus on the *what* over the *why*
- Customer expectations
- Lack of shared vision
- Assumed consensus and misunderstandings

These challenges may be divided into two major categories in terms of what type of solution they should require: practice related, or communication-related. Four of the challenges identified I would like to consider practice related, and the remaining three I regard as being related first and foremost to communication (see Figure 8). The practice-related challenges regard the habit of ensuring pre-RS traceability, division of responsibility, the documentation of rationale, and the tendency to focus on the *what* over the *why*. These four challenges represent practice-related areas that, according to the empirical evidence, were lacking in terms of enabling decision traceability in the projects included in the case study. The practice related challenges are challenges that, in my opinion, practitioners should be able to handle within their projects in two ways. By improving or implementing practices that better support decision traceability, and by working on team engagement and involvement.
The three communication-related challenges are challenges that require improved or new ways of communication with the customer. The communication-related challenges are customer expectations, lack of shared vision, and assumed consensus and misunderstandings.

Note, however, that the division of the identified challenges into the two categories (practice-related and communication-related) is an analytical division. In reality, I assume these challenges to be rather interconnected and dependent on each other, as discussed in the previous chapter. The division into these categories is thus not to say that they can be treated separately from each other. Instead it helps us understand that what type of effort each challenge should require in addressing it.

6.1.2. Important aspects of achieving decision traceability in agile projects

The challenges identified in answering the first research question, highlight some important aspects of achieving decision traceability that consequently became part of the answer to the second research question. Other important aspects identified (project particulars and ownership of process improvement activities) were identified as a result of informants themselves highlighting those aspects.

The aspects identified as important in achieving decision traceability in agile projects are:

- Responsibility and team involvement
- Focusing on the why before the what
- Fostering healthy (customer) expectations for decision traceability
- Project particulars
- Clarifying an agenda for requirements elicitation and traceability
- Ownership of process improvement activities

The aspects can be divided into three major categories; team engagement aspects, project-specific aspects, and process improvement aspects (see Figure 9). The team engagement aspects have to do with engaging the team in achieving decision traceability. They are not project-specific in the sense that they, first and foremost, regard a shift of attitude and focus. The project-specific aspects are aspects that relate the specific project, and that may vary between projects. Essentially, they are aspects that need to be revisited every time a project is initiated. Process improvement aspects are aspects of the higher level of the organization. They regard processes and thus answer to achieving decision traceability in a longer time frame within an organization.

6.2. Implications

A conclusion that can be drawn from the discussion of the challenges in the previous chapter is that several of the identified challenges seem to be interconnected and dependent on each other. Exactly what is the cause of what is not something the current thesis neither aimed to nor can draw conclusions about. However, it occurs to me that their potential interconnectedness may be important to consider when attempting to tackle these challenges. For example, if attitudes about the need to document rationale are what’s causing the lack of rationale documentation, then merely defining practices for documenting rationale will most likely not suffice in achieving decision traceability. With that said, the value perception problem may very well be the cause of all of the identified challenges, in which case value perception is the first challenge to tackle in achieving decision traceability.
The important aspects identified in answer to the second research question have implications regarding what it may take to achieve decision traceability. The types of aspects identified suggest that it takes more than defining processes for achieving decision traceability. Team engagement and a shift of focus when handling requirements is also needed in order to make decision traceability a natural part of the day-to-day work in agile projects. Project-specific aspects highlight the importance of agility in implementing practices for ensuring decision traceability and remind us that the team has a role to play in setting customer expectations that will create opportunities for achieving decision traceability. The process improvement aspects highlight the role of the organization in supporting teams on their quest for decision traceability.

Lastly, I would like to address the implications that the findings of this thesis have on agility. The lack of documentation of rationale is a challenge that may, to some, simply seem like a consequence of agile. The agile manifesto (Beck et al., 2001) does indeed highlight the importance of working software over comprehensive documentation. However, it does not say that having no documentation is the way to go. Agile emphasizes flexibility and empowering the team. My interpretation is that flexibility also means being flexible in the way agile practices are applied, to make it work for the team and the project. If documentation of rationale is something that will empower the team by letting them form a shared understanding of where the project is headed and why, then I would argue that documenting rationale is perfectly justified. In fact, the projects in the case study did already document important decisions. This is something that was appreciated by informants. However, it was the why behind the decisions that didn’t get documented. In my opinion, including the rationale behind the decisions in the documentation shouldn’t take away more agility than documenting the decisions themselves.

Overall the findings of this thesis do not suggest to me that agility must be sacrificed in the name of decision traceability. In my opinion, the findings first and foremost imply the need for a shift of attitude as to who can be held accountable for traceability. They call for attention to the importance of a shared understanding of the motivations behind a project. It is about ownership of the decisions and trade-offs made during the project. It is my belief that the intention to achieve decision traceability will create opportunities for the team to better understand the customer and their problem space. Because the goal of decision traceability is understanding, not documentation. I believe that there can be ways to achieve decision traceability while still maintaining agility, because it depends on customer collaboration and interactions, rather than documentation. The focus should be asking the right questions and enough questions, not overcomplicating the processes and documenting every aspect of a project.

6.3. Knowledge contribution

The knowledge contribution made by this thesis is that of bringing the aspect of the rationale behind requirements and decisions made in agile projects into the discussion of traceability. The empirical case study examined what challenges exist and what important aspects need to be considered in enabling the decisions made during a project – and the reasons for them – to be made traceable. The findings presented in this thesis provide an example of challenges faced by practitioners in real projects, with real customers, when it comes to achieving decision traceability. The important aspects identified contribute to the discussion by proposing factors that, when thoroughly considered, should help organizations conducting agile software projects for customers in achieving decision traceability.
In the research community, the present thesis should contribute to an increased understanding of the topic of decision traceability. Furthermore, this thesis provides a conceptualization of the problem of achieving traceability in relation to product goals and changing requirements. At the most fundamental level, the empirical results of the present thesis confirm the importance of decision traceability in agile software projects.

The discussion on the suspected interconnectedness of challenges and important aspects should also contribute to the understanding of decision traceability as a non-trivial and potentially rather complex problem to solve. One that should need the engagement of everyone involved in a project in order to tackle all of its challenges.
7. Reflection and future research

In this chapter, the methods used and the credibility and application of findings of the present thesis are discussed. Topics and research questions that become interesting for future studies to investigate as a result of the findings from the present thesis are discussed.

7.1. Credibility

Klein & and Myers (1999) defined seven principles for evaluating interpretive IS research. As mentioned in chapter 2, these principles are the ethical considerations I have applied in ensuring and evaluating the credibility of findings in the present thesis. Since I have already discussed my interpretation and application of the principles in section 2.4.1, I will not try to cover every possible aspect of the principles here. However, a couple of the principles highlight methodological aspects of the case study that I believe might affect the credibility of the findings in an unfavorable way. These are the principle of multiple interpretations and the principle of dialogical reasoning.

The most obvious principle where the case study is lacking is that of multiple interpretations. The stakeholder that is most likely to have a different perspective than the informants interviewed, and who would have provided additional interpretations to the importance of decision traceability, is the customer. The fact that the customer perspective is not represented in the empirical evidence does not mean that the customer perspective is not important in achieving decision traceability. Quite the opposite; The conclusions drawn from the case study give the customer a rather significant role in the communication-related challenges and the project-specific aspects. I have also discussed the challenges with customer communication as factors influencing some of the other challenges faced. In hindsight, I believe the customer perspective to be of great significance in understanding the importance of decision traceability. Therefore, the lack of the customer perspective in the present study means that thought the findings may still be useful, there may be alternative interpretations to the challenges discussed. Having access to these interpretations – by interviewing the customer – might have led to a more nuanced analysis and greater credibility of the present thesis.

Another aspect that relates to the principle of multiple interpretations is that since the interviews took place during the course of a few weeks, things happened in the projects alongside the data collection. This meant that a specific issue that P5 and P6 discussed in their interviews was not yet a known issue when the other informants were interviewed. This means that their interpretations of the issue were not recorded. This can be seen as problematic. However, since this was a qualitative study, statistical significance is not an aim. Therefore, one might argue that the interpretations recorded from P5 and P6 do shed light on the issue and that they do have meaning whether the other informants agree with them or not. Therefore, the interpretations given by P5 and P6 should not be disqualified simply because the other informants didn’t get the chance to share their thoughts on one specific issue. However, one conclusion we need to make is that we do not know what the other informants’ take on the matter is. After all, this problem is a consequence of using a case study as the researcher is not in control of the unfolding events. However, the fact that a case study was used may also be of value in considering this issue as I might not have been aware that the issue discussed by P5 and P6 had not happened until after the previous interviews had taken place. In other words, the case study allowed me to gain a deeper insight into the projects, enabling me to understand better the issues and circumstances discussed in the interviews. I will also argue that whether or not the
interpretations of all informants were similar on one specific incident was not central to the purpose and research questions of the present thesis.

In considering the principle of multiple interpretations, the level of structure to the interviews also becomes relevant. This is because the focus of the semi-structured interviews was not to strictly stick to the interview guides, but rather use them as a prompt to start a discussion with informants. Therefore, some themes used in the analysis regard topics that were never discussed in some interviews. This is the result of letting the informants bring up and discuss ideas that they thought to be of importance, but that were not part of the interview guide. In my opinion, this type of loosely structured interview allowed for richer insight into the informants’ interpretation of their reality. The consequence, however, is that all topics were not discussed with all informants.

Another principle that ought to be reflected upon in evaluating the credibility of findings is the principle of dialogical reasoning. The reason for this is the fact that the literature review started before the interviews took place. This is relevant because I, as a researcher, had encountered concepts in the literature review that could potentially be interesting in the analysis. This means that, even though these concepts were not explicitly mentioned in the interviews, the ongoing analysis that is required in conducting semi-structured interviews (in order to ask relevant follow-up questions), might have been influenced by the knowledge gained from the literature review. In this sense, the literature review became the preconceptions I as a researcher brought with me into the interviews. This may have influenced the analysis of findings by increasing the risk of confirmation bias during interviews. However, in order to account for this risk, I have put considerable effort into providing transparency in the presentation of empirical evidence and the reasoning behind my interpretation of empirical evidence. This is why I included a large number of quotes in the presentation of empirical findings and have tried to give extensive explanations for my interpretations of them in the discussion chapter. By providing this transparency, the larger academic community are better able to critically assess the credibility of the present thesis.

Lastly, the principle of contextualization is a principle that could have been applied more heavily by presenting more details of CompanyX and the two projects. However, in order to anonymize CompanyX, the decision was made not to expose more details of the organization’s history and the two projects.

In conclusion, I believe it is important to discuss these principles in hindsight. Simply stating that widely recognized principles for evaluating interpretive research have been applied does not shield my work from critical review and from potential skepticism towards the findings and conclusions presented in my research. Considering the principles does not guarantee credibility any more than documenting information guarantees its traceability. However, the principles do highlight important aspects to consider in conducting interpretive studies. Klein and Myers (1999) note themselves that the principles are not simply rules to follow and that many of them require considerable creative thought in their application. By discussing the principles in chapter 2, I have provided insight into my interpretations and application of them, giving some insight as to my thought process and approach in the research. This makes it possible for the reader to critically assess and judge my trustworthiness as a researcher and the credibility of this thesis.
7.2. Applicability

A couple of aspects may have importance for the applicability of the findings from the case study. The first one is the size of the empirical case study. Relatively few interviews were conducted, and as mentioned in the previous section, the customer perspective was not represented. More interviews would have provided a stronger foundation of evidence for the conclusions. However, it is my belief that the results from the case study may still be of value for more organizations than the one studied. The fact that theory and previous research within the field correlate well to empirical evidence of the case study confirm that the challenges and important aspects identify may be general issues faced in the industry. Furthermore, as discussed in chapter 2, the case study may very well be based on only a few interviews and still yield interesting results. In terms of generalizability, the findings from this study may be analytically generalized as the findings may be abstracted to fit the wider field of software project management.

The two projects that the case study focused on were rather small, which means that the findings may not apply as well for larger projects. The fact that CompanyX was a consultancy firm and that the projects studied were in-house projects, means that the findings are more likely to apply to similar contexts. Furthermore, the different accounts given by informants on how agile the projects really were, makes the focus of the present thesis on agile projects a bit complicated or vague. How well the findings apply to completely agile projects (if such projects exist) – or simply more agile projects – is thus unclear. However, I believe that considering the important aspects of achieving decision traceability identified may be valuable to improving traceability in any organization engaged in agile software development projects.

7.3. Future research

After reflecting on the shortcomings of the present thesis, one may naturally want to consider what future research can examine in order to develop the knowledge on decision traceability further. The first thing that comes to mind to me personally is the perspective of the customer. The present thesis fails to answer to the importance and challenges of decision traceability from a customer point of view, and the findings suggest that the customer has a rather large role to play in achieving it. Therefore, it should be interesting to a study on the customer perspective decision traceability in agile projects. With the present thesis as a starting point, such a study could research one of the following questions:

- How do customers of agile software projects perceive the importance of decision traceability?
- How do customers of agile software projects perceive the challenges of decision traceability?

Another interesting next step in researching decision traceability would be researching how the important aspects identified in the present study may be dealt with in practice. A potential research question for such a study to answer could be:

- What practices can be used in order to achieve decision traceability?

A lesson I have learned from writing this thesis is that traceability is a rather slippery concept. Who determines if an adequate level of traceability exists? The presence of traceability is in some sense in the eye of the beholder. Achieving decision traceability is thus not something we do once and are done. It is not either something that can be easily verified. Answering the
question of how we may achieve decision traceability and what practices should be appropriate, I believe will take more than one study. If I were to attempt to answer the research question proposed above, I would start with a rather extensive literature study examining different practices for traceability. The next step would then be to add the perspective added in this thesis. Namely, the aspect of product goals in relation to changing requirements. This would take us to the next research question:

- How do common practices for traceability enable the traceability of changing requirements in relation to product goals?

Future studies may also investigate how decision traceability may be achieved using agile practices such as Disciplined Agile Delivery and DevOps, as used within the studied organization.

- How can decision traceability be achieved using DevOps?
- How can decision traceability be achieved using Disciplined Agile Delivery?

Another way to approach this topic would be to compare different agile approaches in evaluating how well they enable decision traceability. This would, of course, also require defining how decision traceability can be evaluated.

Considering the specific process model used in the studied organization, a more in-depth case study of how the product goals, initial requirements, and rationale are communicated in agile projects from the start to the end of a project would be very interesting. Such a study could give a more accurate account of how things are actually done in practice, which can give further insight into the challenges of decision traceability. Such a study would, however, have to be conducted over a significantly longer period of time than the case study conducted for the present thesis.
References


Appendix 1. Interview guide (1)

Opening

- Information about voice recording and the right to interrupt the interview
- Verbal consent

Introductory questions

- How long have you worked here?
- What is your role?

Opening questions

- How would you describe the usual course of project?
- What would you say characterizes an agile project?
  - What value does that have?
- How do you apply agile in the projects you are/have been involved in?

In-depth questions

Change

- What type of changes can occur during a project?
- How does it usually go then?
  - How do they surface?
  - Where do they come from?
- How is such a change handled?
- How do you confirm the effect of such a change on...
  - Other parts of the system?
  - Product goals?
- What happens after a decision has been made about such a change?
  - Is it documented?
  - Does it need to be followed-up on in any way? – why/why not?
  - Who is responsible in that case?

Changes and scope

- How can a project's scope be affected by changes?
- (When a change to requirements or features arise in a project, how do you assure that you stay within the scope of the project?)
- Have you experienced having the scope or product goals being revised under the course of a project?
  - If not, how do you think it would need to be handled?
  - If so, how was it handled then?
    - What was the reason?
    - How was it resolved?

Traceability

- What gets traced in a project?
- What do you feel needs to get traced in a project and in that case what kind of tracing?
- How are requirements gathered?
When are they revised?
How are they revised?

Processes and tools

• What tools are used in a project (digital as well as physical)?
  o How are they used?
  o What purpose and what value do they have?
• What kinds of meeting occurs?
  o What purpose and value does each kind of meeting have?

Closing questions

• Summary of what has been talked about
• Do you have anything that you would like to add?
Appendix 2. Interview guide (2)

Opening

- Information about voice recording and the right to interrupt the interview
- Verbal consent

Introductory questions

- How long have you worked here?
- What is your role?

Questions

- What is/has been your role in internal processes at CompanyX?
- How do you perceive the possibility to trace decisions in a project?
- How do you perceive the possibility to trace changes to requirements in a project?
- How do you perceive the possibility to get an overview of how decisions about changes to requirements relate to product goals in a project?
- How does this relate to agility?

Closing questions

- Summary of what has been talked about
- Do you have anything that you would like to add?
Appendix 3. Interview guide (3)

Opening

- Information about voice recording and the right to interrupt the interview
- Verbal consent

Introductory questions

- How long have you worked here?
- What is your role?

Questions

Practices

- What practices do you apply in the projects you are/have been involved in?

Requirements

- How are requirements specified for a project?
- How do you keep track of how requirements relate to product goals?
- Is it documented in any way where requirements come from?
  - Or any motivation for why each requirement exists?

Change

- How do you handle decisions that regard changes to the specified requirements?
- In the cases that requirements get changed, how can you follow-up in order to confirm that product goals are still fulfilled by requirements?

Traceability

- What do you feel needs to get traced in a project and in that case what kind of tracing?

Closing questions

- Summary of what has been talked about
- Do you have anything that you would like to add?
Appendix 4. Interview guide (4)

Opening

- Information about voice recording and the right to interrupt the interview
- Verbal consent

Introductory questions

- How long have you worked here?
- What is your role?

Questions

Practices

- What practices do you apply in the projects you are/have been involved in?

Requirements

- What is your perception of how requirements are handled in the projects you have joined now?
- How do you keep track of how requirements relate to product goals?
- Is it documented in any way where requirements come from?
  - Or any motivation for why each requirement exists?

Change

- The recent changes that have arisen in ProjectX, what is your perception of what has happened?
  - What could CompanyX have done differently in this case?
  - What needs to get done now?
- I de fall då krav ändras, hur kan man följa upp för att säkerställa att produktmål fortfarande uppfylls av kraven?
- In the cases that requirements get changed, how can you follow-up in order to confirm that product goals are still fulfilled by requirements?

Traceability

- What do you feel needs to get traced in a project and in that case what kind of tracing?

Closing questions

- Summary of what has been talked about
- Do you have anything that you would like to add?