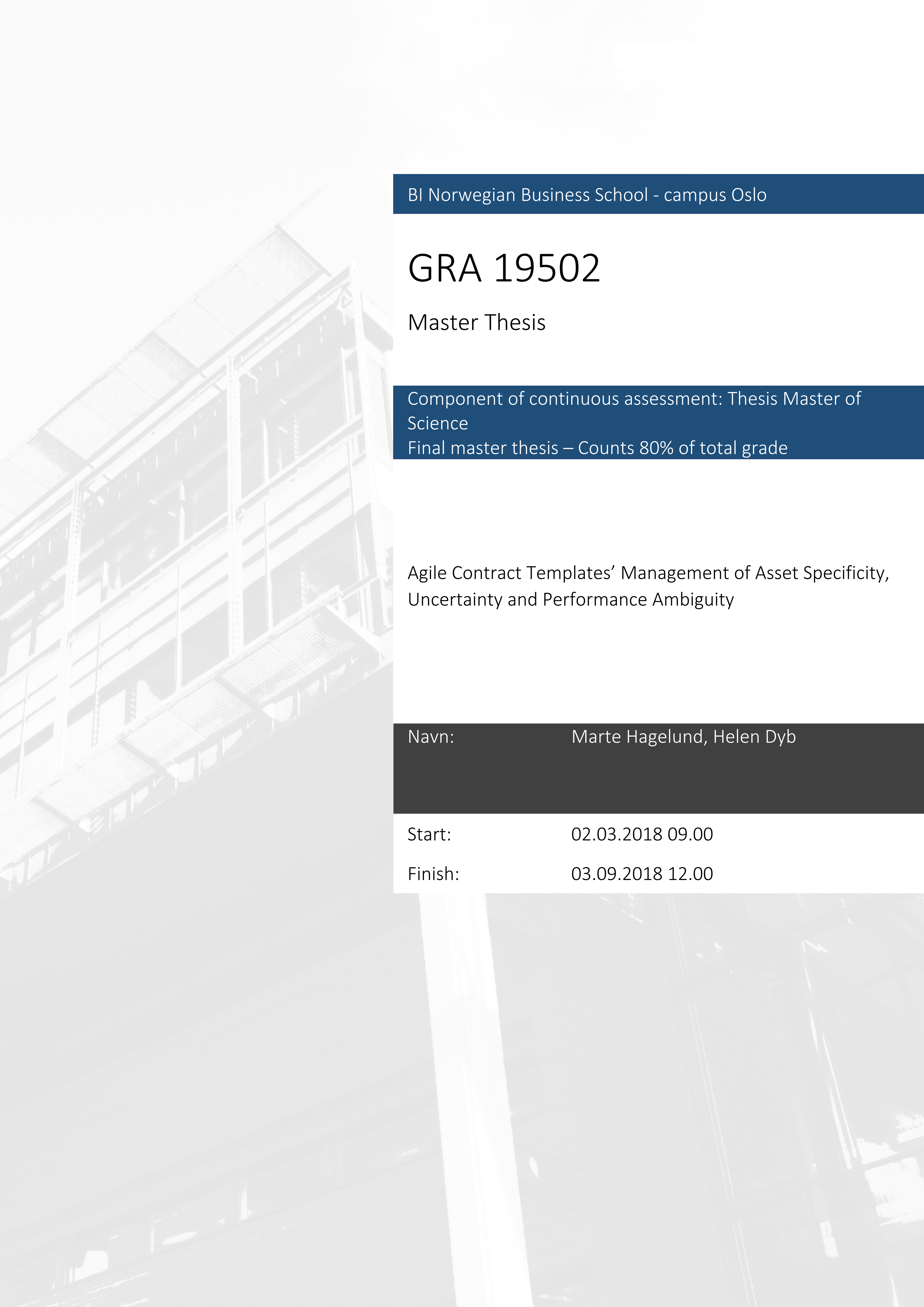


BI Norwegian Business School - campus Oslo

GRA 19502

Master Thesis

Component of continuous assessment: Thesis Master of Science
Final master thesis – Counts 80% of total grade

Agile Contract Templates' Management of Asset Specificity, Uncertainty and Performance Ambiguity

Navn:	Marte Hagelund, Helen Dyb
Start:	02.03.2018 09.00
Finish:	03.09.2018 12.00

BI Norwegian Business School

Examination code and name:
GRA 19502
Master Thesis

Agile Contract Templates' Management of Asset Specificity, Uncertainty and Performance Ambiguity

Date of submission:
01.09.2018

Campus:
BI Oslo

Programme:
Master of Science in Strategic Marketing Management

Name of students:
Helen Dyb & Marte Hagelund

"This thesis is a part of the MSc programme at BI Norwegian Business School. The school takes no responsibility for the methods used, results found and conclusions drawn."

Executive Summary

The majority of today's firms must take a stand to Industry 4.0, and how they can digitalize their business. Large investments in ICT projects are therefore made by customers of technological solutions, but reports find that there are numerous examples of budgetary transgressions and other factors that contributes to failed projects. As a response to the need for better methods in ICT development projects, the Agile method for software development emerged, aiming for end-solutions to be developed during a project in order to satisfy the customer by creating a better value outcome. Further, Transaction Cost Economics (TCE) note that there are several transaction hazards that can affect a buyer-supplier-relationship, thus different governance mechanisms to cope with these.

The aim of the study is to answer the research question:
How does agile contract templates cope with transaction hazards in ICT projects, and how does that affect when to use the contracts?

Linked to the research question are three transaction hazards found in TCE, which we divide in three sub-questions:

1. How does the agile contract templates cope with asset specificity?
2. How does the agile contract templates cope with uncertainty?
3. How does the agile contract templates cope with performance ambiguity?

The research question is answered with a qualitative research method, where the data sources consist of agile contract templates and interviews. The result of the study finds that the agile contract templates differ in their degree of enabling software development based on the Agile method, which in turn effect how the templates cope with the transaction hazards. However, independent of which agile contract template being used, we find that good project management, relational governance and high customer competency cope with the three transaction hazards to the highest degree. Further, we find that the customer who desires a more agile project and is able to bear the responsibility of risk elements should choose SSA-S, whereas the customer with higher risk-aversion and requirements of a rigid framework should choose PS2000 Agile.

Acknowledgement

This master thesis marks the end of two educational and demanding years at BI Oslo as students in the program of “Strategic Marketing Management”. We can therefore proudly state that we have completed our degrees and are now looking forward to applying the knowledge in the next chapter of our lives.

Our study concerns governance mechanisms and agile method in ICT projects, a topic we believe will increase in relevance due to Industry 4.0.
In addition to the appeal of digitalization, we find strategy and management interesting, and especially why some businesses succeed whilst others do not. One way of succeeding is by staying ahead of competitors and another is to safeguard your operations and thereby maximize the value output, which have served as inspiration for the choice of topic.

Many “thank you” are in order for the result of this thesis. First and foremost, we want to thank our supervisor, Jon Bingen Sande, for your time, effort and positive attitude towards us and our thesis. All your input, knowledge and constructive feedback have contributed to a result that we are proud to present. Secondly, we want to thank all our informants, who took the time to participate in interviews and provided us with invaluable knowledge and experiences. Without your contribution we would not be able to answer the research question in the same extent.

We also want to thank our family and friends for support, positive comments and encouragement throughout this long process.

Lastly, we want to thank and congratulate each other. The process of completing this thesis has been challenging at times, but above all highly educational.
We have had numerous meetings, some discussions, several laughs, but above all a great collaboration, which has led to the thesis you are now holding.

We sincerely hope you will enjoy reading our master thesis!

Oslo, 01.09.2018
Marte Hagelund & Helen Dyb

Table of Content

Executive Summary ... 1

Acknowledgement ... 2

1. Introduction ... 6

1.1 Research topic ... 8

1.2 Structure of the Master Thesis ... 9

2. Literature Review ... 10

2.1 The Agile Method ... 10

2.1.1 The Agile Method in Development Processes ... 11

2.1.2 Iterative development ... 12

2.1.3 Waterfall development ... 12

2.2 Factors Affecting ICT projects ... 13

2.2.1 Success Factors ... 14

2.2.2 Factors with Negative Impact on Success ... 15

2.3 Conclusion of the Agile method and Factors Affecting ICT projects ... 16

2.4 Transaction Cost Economics ... 16

2.4.1 Transaction Attributes ... 16

2.4.1.1 Asset Specificity ... 17

2.4.1.2 Uncertainty ... 18

2.4.1.3 Performance Ambiguity ... 18

2.4.2 Governance Forms in Transaction Cost Economics ... 19

2.4.2.1 Market Governance ... 20

2.4.2.2 Hierarchical Governance ... 20

2.4.2.3 Relational Governance ... 21

2.4.3 Formal Contracts and Relational Governance as Complements ... 21

2.4.4 Misaligned Contracts ... 22

2.4.5 Contracts in Innovating Industries ... 22

2.5 Conclusion of Literature Review ... 23

3. Research Method ... 25

3.1 Choice of Method 25

3.2 Qualitative Method 25

3.3 Data Sources 26

3.3.1 Data source 1: Contract templates 26

3.3.1.1 Contract template 1: Difi's SSA-S 27

3.3.1.2 Contract template 2: DND's PS2000 Agile 28

3.3.2 Data Source 2: Qualitative Interview 29

3.3.2.1 Method of Interviews: In-depth & Semi-structured Interviews 29

3.3.2.2 Sampling of Informants 30

3.3.2.3 Interview guide 32

3.3.2.4 The Interviews 33

3.4 Qualitative Data Analysis 34

3.5 Quality in Qualitative Research 37

3.5.1 Validity 37

3.5.2 Reliability 39

3.5.3 Critical Assessment of Informants 40

3.5.4 Ethical and Legal Responsibility 41

4. Analysis of the Agile Contract Templates 43

4.1 Explanation of Terms 43

4.2 Analysis of the Relevant themes in the Contract Templates 44

4.2.1 Need Analysis 45

4.2.2 Solution Proposal and Scope 46

4.2.3 Risk Assessment 49

4.2.4 Price Model 53

4.2.5 Project Group 56

4.2.6 Delivery Plan and Competency 59

4.2.7 Termination 67

4.3 Transaction Hazards 70

4.3.1 Asset Specificity 70

4.3.2 Uncertainty 71

4.3.3 Performance Ambiguity 72

5. Discussion 75

5.1 The Agile method versus Agile Contract Templates.................................. 75

5.2 The Agile Contract Templates versus Coping with Transaction Hazards... 76

5.3 Managerial Implications and Decisions 82

5.3.1 Managerial Implications 82

5.3.2 Managerial Decisions.. 83

5.4 Limitations & Future research 84

5.4.1 Limitations 84

5.4.2 Future Research 86

6. Conclusion 88

7. List of Figures and Table 89

7.1 List of Figures... 89

7.2 List of Tables 89

8. Appendices 90

Appendix 1: Agile Manifesto 90

Appendix 2: Interview Guides... 91

Appendix 3: Informant Request 95

Appendix 4: Quotation-check.. 96

Appendix 5: Analysis Process 97

Appendix 6: The Risk Matrix... 100

9. References... 101

1. Introduction

Since the 18th century, the world has gone through different industrial revolutions that significantly impact peoples' way of living. As of today, we find ourselves in the 4th Industrial Revolution (Harash, 2016) called Industry 4.0, where technology enables businesses to enhance their delivery, growth and productivity, and transform products and supply chains (ForbesInsights, 2018; Cotteleer & Sniderman, 2017). Thus, large opportunities are available for firms to gain competitive advantage by investing in technology.

In 2010, The Norwegian Information Technology (IT) industry had a turnover of 202 billion NOK (Regjeringen, 2013), and leading companies state that digitalization is a priority on their agenda (Kommunal-og moderniseringsdepartementet, 2015; Marschall & Korstvedt, 2017).
Gartner (2017) defines digitalization as "*...the use of digital technologies to change a business model and provide new revenue and value-producing opportunities; it is the process of moving to a digital business".*

Despite the large investments, a report by Jørgensen (2015) finds a substantial part of the Norwegian Information and Communications Technology (ICT) projects are delivered with small benefits and/or great budget transgressions. Although the latter is often covered by the media as the greatest problem related to ICT projects, this is however not the best indicator of whether the projects are successful. Other factors contribute heavily, and research show that success can be linked to elements such as communication, processes handling, changes and training, processes for project and risk management and more (Jørgensen, 2015).

In order to cope with the above-mentioned elements and complexity of ICT projects, research has tried to find ways to optimize projects. Emerging from these studies is the observation that agile methods lessen the budgetary transgression, enhances the quality and increases predictability of the projects (Larman, 2004). Because of these results, the method has gotten a foothold in the Norwegian IT industry (Brevik & Grønli, 2013).

Other elements that can challenge the success of a project are covered by Transaction Cost Economics, where the risk of stakeholders acting opportunistic to safeguard own investments is central (Williamson O. , 1979). Such behavior cause problems in the relationship between a customer and supplier and can further contribute to ICT projects' risk and failures. Appropriate governance mechanisms are therefore considered a key managerial decision as they play an important role in inter-firm relations, by regulating the framework for exchange between the involved parties (Ghosh & John, 1999; Sande & Haugland, 2015). Governance mechanisms are found among other things in contracts, thus making successful projects dependent upon the formation of these and their enablement of agile project execution. Hence, the appropriate use of governance mechanisms which here entails the use of agile contracts, can affect hazards in ICT projects as it lays the foundation for co-operation and successful technological development.

The purpose of this thesis is to contribute to a better understanding of how the use of different agile contracts from Norwegian institutions cope with transaction hazards, and how this affects the use of different contracts. Through our findings we are able to provide greater insight to how agile contracts can contribute to successful completion of agile ICT projects.

1.1 Research Topic

Due to the trends emerging from Industry 4.0, reports show that companies both prioritize and invest in digitalization, where agile methods have gotten its foothold in software development projects. ICT projects are as other projects also affected by challenges raised by Transaction Cost Economics, but despite the projects' high failure rates, little research has been done to understand if contracts enabling agile method copes with the challenges presented in this theory.

Thus, our study aims to provide a better understanding of how contracts enabling agile method affect ICT projects through mechanisms coping with transaction hazards, and how this in turn affects the choice of appropriate contract.

The research question of this master thesis is the following:

How does agile contract templates cope with transaction hazards in ICT projects, and how does that affect when to use the contracts?

Sub-questions

In order to investigate the research question more in depth, we have decided to consider three transaction hazards found in TCE as three sub-questions:

1. How does the agile contract templates cope with asset specificity?
2. How does the agile contract templates cope with uncertainty?
3. How does the agile contract templates cope with performance ambiguity?

1.2 Structure of the Master Thesis

This master thesis consists of six chapters that collectively aim to give the reader a thorough and deeper insight to the research question previously described. The last three chapters consists of appendices, list of figures and tables, and references.

Some quotes are found in Norwegian books and/or articles, thus translated to English. This is symbolized with * behind the quote. e.g.: "*social phenomena are complex, and the qualitative interviews makes it possible to enlighten the complexity and nuances**".

- **Chapter 2 Theoretical Frame of Reference:**
  The second chapter presents the Agile method, and theory regarding Transaction Cost Economics and governance mechanisms.
- **Chapter 3 Method:**
  The third chapter demonstrates how we collected and analyzed the data used. This chapter enlightens the theory behind qualitative methods, and how this is applied in this study.
- **Chapter 4 Analysis agile Contract Templates:**
  The fourth chapter provides an analysis based on our data sources, where we have extracted seven themes of focus. Conclusion on the three transaction hazards will also be presented.
- **Chapter 5 Discussion:**
  The fifth chapter builds on the analysis from the fourth chapter, where we discuss findings concerning the research question, and managerial implications and decisions. Also, limitations and future research are presented.
- **Chapter 6 Conclusion:**
  The sixth chapter consists of a short conclusion of the study and the research question.
- **Chapter 7 List of Figures and Tables**
- **Chapter 8 Appendices**
- **Chapter 9 References**

2. Literature Review

In this chapter of the master thesis we elaborate on the Agile method, its origin and processes, and how it is used in projects. Further, we look into reports concerning factors affecting the success and failure of ICT projects.
Lastly, related to ICT projects is the transaction between the supplier and buyer, and we therefore consider Transaction Cost Economics, and different governance mechanisms.

2.1 The Agile Method

In its early days, software development suffered from slow evolvement, long lead time and difficulties regarding changes, and therefore business leaders and developers sought out ways of developing software that did not require extensive controlling and rigid processes. Instead, methodologies including fast delivery approach and ability to respond to change was desired (Koch, 2005). Thus, the Agile method emerged.

The Agile Alliance was established in 2001, where 17 method experts formulated "The Agile Manifesto" in Utah, USA (Beck, et al., 2001).
It is built on 4 core values and 12 principles, which form the basis for agile software development methods.
The figure below illustrates the 4 core values, which is the foundation of the Agile Manifesto:

Manifesto for Agile Software Development

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

That is, while there is value in the items on the right,
we value the items on the left more.

Figure 1: Manifesto for Agile Software Development

The 12 principles are viewed as guidelines for using the Agile method, where change is welcomed, and the customer is the main focus.
The first principle of the manifesto reflects the highest priority*: satisfying customers by delivering valuable software*. This aligns with studies finding valuable outcome as a critical factor of software development (Alahyari, Svensson, & Gorschek, 2016). The 12 principles can be found in Appendix 1.

We refer to the Agile method when the method is compliant to the Agile Manifesto, meaning in its true form. However, the Agile method often serves as a base, where projects are agile but not to the full extent of the Agile Manifesto, thus we refer to this as agile method.

2.1.1 The Agile Method in Development Processes

The method enables developers to do changes in the software specifications as the project moves along, and by that challenge rigid formal contracts as the method require flexible conditions (Abrahamsson, Salo, Ronkainen, & Warsta, 2002). Experts in the field recognize that there will not be a one-size standard suitable for all deliveries (Abrahamsson, Salo, Ronkainen, & Warsta, 2002). This means that based on the project and the development, the parties must find how agile the project should be.

The Agile method can be difficult to understand if one has not heard of it before. We therefore provide an example that hopefully enables the reader to better grasp the main point behind the method:

Imagine getting into a car where you need to travel to a distinct destination, where the destination serves as a metaphor for the end-product. Following traditional methods (waterfall), you will use a map with pre-specified directions, and you will reach the specific destination eventually. With the Agile method, you get in the car, but without a map. You will still have a destination to reach, but the idea is that with the Agile method enables the driver to find new roads as it is moving along, and these roads might even be better than the ones pre-specified in a map. Thus, this can lead to a better destination. Additionally, as the car moves along, one can go back and forth until the best possible road is found. This is supposed to lead to the best possible outcome, and not necessarily the pre-defined destination. *(Example given to us by Mari Vestre at Difi, who has created an agile contract template).*

To understand the difference of implementing the Agile method versus traditional development methods, we provide an explanation of the crucial element in the Agile method called Iterative development in the following section, followed by explaining its counterpart, called Waterfall development.

2.1.2 Iterative development

Iterative development consists of series with iterations where one refers to a repetition, or the re-work, of current activities. Iterations in development processes break down larger developments into smaller bits and can be loops or cycles of feedback (Yang, Lu, Yao, & Zhang, 2014). As the project is developed and tested in repeated cycles, additional features can be designed, developed and tested until full functionality. Therefore, iterations enable the project to become a series of smaller self-contained projects of short cycles within pre-decided time limits, with dependency upon results from the previous iteration (Spence & Bittner, 2005). Hence, agile methods enable software solutions to be developed and delivered in a much shorter time period than what earlier methods have, e.g. Waterfall model (Miller, 2001).

With high failure rates in ICT projects, often related to specification problems and unclear end-results, iterations have become important due to the need for testing and failing. This is clearly specified through the statement that *"Agile software processes acknowledge that we get things wrong before we get them right"* (Miller, 2001, p. 1).

2.1.3 Waterfall development

Iterative development stands in contrast to the traditional Waterfall model where each phase is "closed" before moving to the next (Jonnalagadda, et al., 2017). The Waterfall model is a well-known process for project management and system development and is in its simplest form based on the implication of linearity and sequentially. Thus, it consists of separate phases, similar to a downward ladder. One step must be completed before the next one is started, and the steps cannot be revisited, hence leading to a high degree of rigidness (Pries & Quigley, 2010; Cobb, 2015).

The figure below illustrates the differences between Waterfall and Iterative development (Moniruzzaman & Hossain, 2013) :

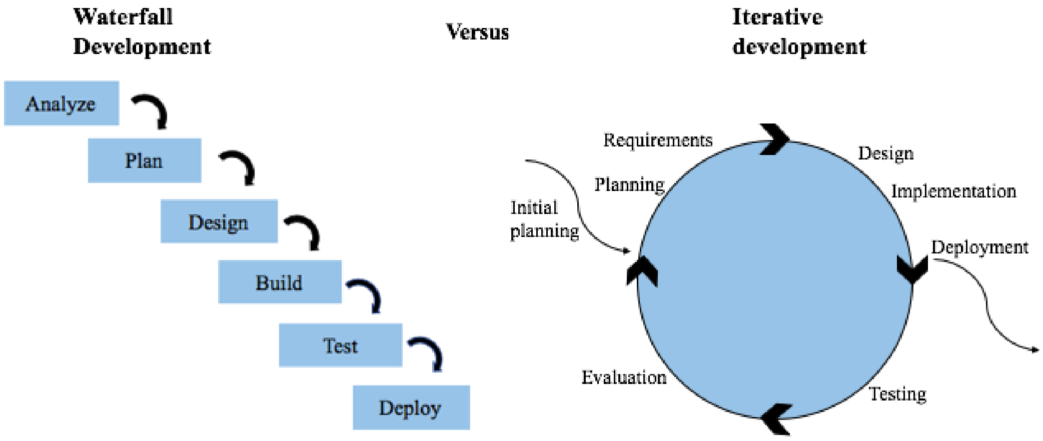


Figure 2: Illustration of differences between Waterfall development and Iterative development

2.2 Factors Affecting ICT projects

In the following section and sub-sections, we consider the findings from studies that demonstrates factors affecting the success of Norwegian ICT projects.

Reports show that several billion NOK are invested each year for development and management of ICT solutions (Kommunal-og moderniseringsdepartementet, 2015; Marschall & Korstvedt, 2017). This demonstrates that the Norwegian business society focuses on digitalization and believes in its benefits. Despite the large investments, several ICT projects are delivered with great budgetary transgression. Public sector shows an average of exceeding budget with more than

30 per cent over a longer period of time, and where 10 per cent of the investments ends with no delivery at all (Jørgensen, 2015). Moreover, 30 per cent of the ICT projects also fail in terms of delivering less benefit and value than primarily indicated and demanded (Jørgensen, 2015).

However, there is a low correlation between delivered value and level of budget control (Jørgensen, 2015). Thus, this indicates a willingness to exceed primary budget if it is considered to bring more value than the plan gave, hence one is open for change. The latter is therefore a good fit with the Agile method, where one must be open to widen the scope and by so increase the cost of the development project in order to gain more value.

2.2.1 Success Factors

Succeeding with ICT projects requires great co-ordination and partnership through well-established processes and involved stakeholders from both the customer and the supplier (Patanakul, 2014; Jørgensen, 2015). These success factors align with the Agile method, where the focus on partnership and co-ordination reflects core values 1 and 2 of the Agile Manifesto. Further, the review done by Jørgensen (2015) notes that success factors depend upon whether the contract used incentivizes the supplier, and the use of a risk analysis and a subsequent adjustment of ambitions.

Jørgensen (2015) finds that certain conditions and measures can be done to increase the likelihood of successful projects. One is to split larger projects into smaller deliveries, i.e., having a high frequency of deliveries during the project. Another factor is to obtain considerable analysis of the customer’s need, and follow-up on this during the project’s life-cycle and use it as a base when evaluating deliveries. This corresponds with core value 4 of the Agile Manifesto. In addition, there is a need for comprehensive involvement and competence provided from the customer. This enables development processes that can foster change to the project’s demands and objectives, which in turn can increase the value of the delivery. The customer involvement is defined as important in core value 3 of the Agile Manifesto.

Based on the section above, there seems to be a link between the success factors in ICT projects noted by Jørgensen and the core values in the Agile method. Other reports also demonstrate beneficial correlation between using agile method and successful ICT projects. In 2017, PricewaterhouseCoopers (PWC) released a report showing that projects using agile method were 28 per cent more successful than traditional projects (Jonnalagadda, et al., 2017).

The same report from PwC also states that implementing agility to the development helps IT projects create the minimum viable product, meaning the minimum number of features that will create the valuable solution to the customer. Thus, it avoids additional features that surpasses the end-user's need. This corresponds to the findings by Mannaro, Melis and Marchesi (2004), where IT developers in agile projects were more satisfied with the method as opposite to others. Their reasoning was due to their experience of creating a higher value outcome for the customer by enhanced efficiency and quality in the development process.

2.2.2 Factors with Negative Impact on Success

Jørgensen (2015) notes that negative factors of a project's result are linked to lack of knowledge and understanding of the elements needed in a project. Moreover, the parties can underestimate the complexity and risk of the development, thus projects that with correct knowledge would have been deemed too complex or risky are started.

It can be difficult to evaluate the competence needed in the project, such as assessment of own quality towards processes. This is further shown in problems related to a poor understanding of the project and interaction between the involved parties that negatively contribute to achieve a successful result. In addition, it can be difficult to organize a project leadership that creates the optimal co-operative environment for success.

Lastly, incentives can affect the supplier's behavior, and Jørgensen (2015) emphasize that the use of fixed-price contracts often proves to deliver less value in the reviewed ICT projects. Fixed price contracts require clients to define the solution in advance of the project in order to adequately price the whole scope of work. This price model is not considered to be aligned with the Agile method as this contradicts its main point of having a low degree of defined aspects and an open scope.

2.3 Conclusion of the Agile method and Factors Affecting ICT projects

Based on the previous sections and our findings of correlation between the Agile method's core values and proven success factors, we consider the Agile method to enable processes that ensure a higher success rate of ICT projects.

Because a transaction is the foundation in all ICT projects, the next section considers how these are affected by transaction attributes and governance mechanisms.

2.4 Transaction Cost Economics

From Transaction Cost-theory emerged Transaction Cost Economics (TCE) in order to understand the essence of the classic make-or-buy decision and organizations' choice of governance mechanisms in transactions (Williamson O. E., 1981). In all transactions between suppliers and buyers, the parties must consider the linked transaction attributes and costs, thus structure a governance that will enhance the value and decrease its costs.

A transaction is recognized as to *"...occur when a goods or service is transferred across a technologically separable interface"* (Williamson O.E., 1981, p. 552).

2.4.1 Transaction Attributes

Transaction attributes are an important part of TCE as they can affect the value outcome of the exchange. The attributes will differ depending on the transaction, and lead to various degrees of complexity and hazard, thus one must cope with transaction attributes differently. In the literature concerning TCE, asset specificity, performance ambiguity, frequency, knowledge leakage and uncertainty is commonly considered as important transaction attributes (Williamson O. E., 1981; Poppo & Zenger, 2002; Rindfleisch & Heide, 1997).

From the different transaction attributes, we recognize asset specificity, uncertainty and performance ambiguity to be most frequently considered in TCE. We find these to be important in ICT projects because they require large investments that needs to be protected, where the projects are uncertain due to agile development which also can create performance ambiguity. Below follows an elaboration of the three transaction hazards and how they are relevant to ICT projects.

2.4.1.1 Asset Specificity

Williamson (1981; 1985) notes asset specificity as the most critical transaction attribute as it concerns transaction-specific investments aiming to gain maximum value with minimum cost related to it. Certain transactions involve the need for specific investments in specialized assets from one or both parties, which will be crucial to create value.

The main point with asset specificity is that with its rigid primarily purpose, the investment will lose value if re-deployed for other objectives outside the buyer-seller-relationship (Williamson 1981;1985). Thus, it creates a lock-in effect for both parties if they have invested in the co-operation. However, the strength of the lock-in varies with the level of the concreteness of the asset specificity affecting if it can be used for other purposes.

In agile software development, specific investments can involve how one has allocated specific human resources to the project. Included in human capital specificity is also how both parties may develop specialized competence of knowledge and skills to enhance and complete the project. Furthermore, asset specificity can concern physical assets where investments are done in specialized tooling and equipment that are customized for the specific solution being implemented.

Asset specificity is interlinked with uncertainty, meaning that uncertainty will follow when asset specificity exists.

2.4.1.2 Uncertainty

Uncertainty in TCE concerns the risk of changes in the circumstances of the transaction, such as unanticipated behavior or unpredictable happenings in the environment. (Schepker, Won-Yong, Martynov, & Poppo, 2014; Carson, Mahok, & Wu, 2006).

All companies operate in evolving and changing environments. This means that different events in the market place can affect the parties' businesses, which in turn can affect the progress of the project and therefore create uncertainty.
Also, technology itself is changing and evolving more rapidly than before, which can affect the development and implementation of the software.

Additionally, the development and implementation of the software in ICT projects are often complex processes that involves many stakeholders who have different professional background and objectives (Denning, 2016).
Therefore, with more complexity and uncertainty, the higher the risk is for failure if not addressed appropriately and taken into consideration when creating the contract and building the relationship.

Because of these uncertainties, the Agile method can be employed to enable a framework that is flexible and open to changes during the development (Denning, 2016). Because of uncertainty, performance ambiguity can arise in the delivery.

2.4.1.3 Performance Ambiguity

Ambiguity arises from the uncertainty of the cause-effect relationship in the supplier's delivery (Carson, Mahok, & Wu, 2006). We find performance ambiguity to be relevant in our study as it deals with the delivery in agile development, and the evaluation of whether the delivery is accepted as successful or not. Thus, the customer and supplier must consider success based on the same criteria.

However, even with agreed success criteria, it can be difficult to evaluate a delivery in an objective manner due to different view-points as to what eventually led to success or failure of the result, thus creating performance ambiguity (Yang, Lu, Yao, & Zhang, 2014). This is often the case when the customer states that the supplier has failed in its delivery. In such cases, the customer argues that the solution provided is not what was agreed upon and conclude that success criteria have not been met. The supplier might however argue that the customer was vague with regards to their criteria and demands from the outset, and that they did not provide resources with adequate competence to evaluate the project in time or in a proper manner, therefore affecting the outcome (Yang, Lu, Yao, & Zhang, 2014).

2.4.2 Governance Forms in Transaction Cost Economics

When the parties decide on a development project, they must choose how to structure the governance that mediates the transaction and its consequences, hence safeguard the investments.

Sande and Haugland (2015) notes that a governance structure should be aligned with the transaction attributes, as overly formal contracts can inhibit adaptiveness to the circumstances. However, lack of rigid formality can cause opportunistic behavior, thus conflict (Williamson O. E., 1981). With conflicts in an exchange, Hendrikse (2003) finds that transaction costs arise. Such costs are often linked to the need to re-negotiate and re-write contracts, the search of a new partner, and cases where the value of the parties' investments, such as asset specificity, is being reduced or lost. Hence, an alignment of the governance mechanism and transaction attributes is important.

In TCE, market governance, hierarchic governance and relational governance are referred to as forms for structures (Williamson O. , 1979; Ghosh & John, 1999). The type of governance form is an important decision among the involved stakeholders, and is affected by the firm's resources of technology, end-customers and supply chain, but also the transaction attributes (Ghosh & John, 1999). Accordingly, the choice of structure will greatly impact the firm and the relation between the involved parties.

In the following sub-sections, we present the three governance structures and their implications.

2.4.2.1 Market Governance

In a market governance the objective is to create a standardized contract where the mechanisms in the contract are affected by the events in the market (Ghosh & John, 1999). The meaning of market governance is therefore to create incentives in the contract where the party who is acting opportunist will lose potential future business value. This is also referred to as classical contracting, where formal contracts are most used.

A contract is a created framework that guides exchange between parties (Llewllyn, 1931). It defines the roles and responsibilities divided between the parties, and work as a tool to enforce exchange (Macaulay, 1963). A contractual document forms the natural base for content and scope of the contract (Haaskjold, 2013).

According to Sande and Haugland (2015), formal contracts regulates the relationship between firms by having written and detailed documents which includes role specification and contingency planning. In addition, Jiang et al. (2012) notes that according to TCE, formal contracts also restrain the involved firms' opportunistic behavior by the use of reward/punishment stipulations. Additionally, the contracts are framed by a legal 3rd part (the court of law), which also is critical to protect against opportunistic behavior. Consequently, it follows that an effort to build a relationship between the parties are not conducted (Williamson O. , 1979).

2.4.2.2 Hierarchical Governance

Hierarchical governance is a form of governance mechanism to vertically integrate the exchange to one's business, meaning to take ownership of assets (Ghosh & John, 1999). This governance structure provides full control of the assets as the exchange is internalized and is therefore not as vulnerable to opportunistic behavior as the other two governance forms (market and relational governance). An internalized governance can be seen when a firm needs specialized competence to create superior value, thus one builds the skills in-house and try to avoid imitation (Ghosh & John, 1999).

2.4.2.3 Relational Governance

During the 90s, a shift in the buyer-supplier nature appeared, where relationships became closer through alliances and partnerships (Heide & John, 1990).
Thus, relational governance emerged to capture long-term business relationships, by function as a hybrid of market and hierarchical governance (Burket, Ivens, & Shan, 2012). The uniqueness of relational governance lies in the mutual trust it engenders in a partnership, where both commit to the informal agreement in order to hold a sustainable relational contract (Ghosh & John, 1999; Poppo & Zenger, 2002).

Heide and John (1992) identifies three measures the parties should share in a relationship based on norms: solidarity, flexibility and information exchange. Solidarity considers the expectation that both firms place a high value on the relationship. Additionally, it is expected that the parties are willing to make adaptions, and lastly that they proactively provide important information to the other party. These norms are consistent with the important factors leading to successful ICT projects and aligns with agile method (Jørgensen, 2015).

2.4.*3 Formal Contracts and Relational Governance as Complements*

More recent literature regarding governance forms seems to move away from the traditional idea that formal contracts and relational governance are mutually exclusive. The literature rather presents the idea that the two forms should be taken use of strategically and in combination to achieve higher effectiveness for the firms involved (Sande & Haugland, 2015; Poppo & Zenger, 2002; Gibbons & Henderson, 2012).

Through empirical research, Poppo and Zenger (2002) finds that formal contracts and relational governance function as complements.
The authors found that the two contract forms have different origins, thus serving diverse functions which makes them complement each other. Hence, the use of

one form leads to an increase in use of the other. Their study demonstrate that well-crafted contracts enhance trust in long-term exchanges, where contracts can be customized to fulfill what is mutually agreed upon. To optimize the exchange, relational governance will help cope with uncertainty and need for adaptiveness, and by that help to safeguard against conflicts that is difficult to outdraw in a contract.

2.4.4 Misaligned Contracts

Sande and Haugland (2015) argue that relational contracts mediate the effect of misaligned formal contracting on performance. They point to studies that shows the importance of fit between formal contracts and transaction attributes, and how a misalignment will reduce relationship performance.

Sande and Haugland (2015, p. 188) define misaligned formal contracting as "*The distance between the realized level of formal contracting and the expected or appropriate level of formal contracting under given set of transaction attributes*".

They emphasize the importance of finding the optimal fit of the contract and the transaction attributes to cope with transaction hazards. Their conclusion is however that the impact of a misalignment will be more negative for end-product enhancement, which relies more on relational contracts. This opposite to cost-reduction outcomes, which is more easily written in a formal contract with measurable targets.

2.4.5 Contracts in Innovating Industries

We consider ICT projects to be an important part of innovative development, and below we review the literature linked to such transactions.

Studies of TCE note that firms should aim for hierarchy forms when both asset specificity and uncertainty are high (Riordan & Williamson, 1985; Williamson O. , 1979). However, Gilson, Sabel and Scott (2009) emphasize that in innovating industries, producers are moving towards vertical disintegration due to their dependence on other manufacturers to provide best-in-class technological products.

Due to this shift, Gilson, Sabel and Scott (2009) finds that a new contracting form called "contracting for innovation" has surfaced, where uncertainty in the innovation process is coped with through high switching costs.
They find uncertainty to arise from the inability to specify final products in innovative development projects. We find this to align with using agile method where the end-result should be unknown.
Further, the authors note that switching costs implies that the buyer's ability to change supplier will no longer be beneficial post initial purchase. In collaboration processes, the investment of learning the supplier's products and building the relationship of mutual understanding gets lost if the customer change supplier, and there will be costs searching for a new partner. Additionally, as the project moves forward the investments grow, thus the barrier to switch partner increases, which function as a constraint for opportunistic behavior (Gilson, Sabel, & Scott, 2009).

As a result, Gilson, Sabel and Scott (2009) notes that collaborative processes are replacing vertical integration with focus on relational contracting, where trust can facilitate dealing with uncertainty within a framework creating a lock-in to the co-operation. Establishing collaboration and trust are also important elements when executing an agile project aiming for flexible development open for changes (Abrahamsson, Salo, Ronkainen, & Warsta, 2002). Additionally, Gilson, Sabel and Scott's (2009) findings are aligned with Poppo and Zenger's (2002) conclusion of formal contracts and relational governance being complements.

2.5 Conclusion of Literature Review

From the literature review we recognize success factors of ICT projects to be highly aligned with the Agile method. At the same time, factors that negatively impact the success of ICT projects are found contradictive to the Agile method. Thus, agile method's high usage rate in the market seems appropriate and enables higher value outcome of the projects.

The literature review of TCE demonstrates that asset specificity, uncertainty and performance ambiguity are relevant for agile development projects as they can affect the value outcome of the exchange. We find that the transaction hazards will be present in ICT projects through e.g., large investments, the fast pace of technological change in the market, and the need for customized delivery, thus creating a need for safeguarding and coping mechanisms.

Therefore, choosing the structure of governance mechanism is of key managerial decision to protect a firms’ interests by utilizing the complementary effect of formal contracts and relational governance that aligns with the transaction attributes. We believe this will be affected by implementing an agile method, thus influence the choice of contract template.

3. Research Method

In the following chapter, we present the chosen research method for this study. Theory behind the choice of method is provided, where we explain in detail how we collect, process, and analyze the data.

3.1 Choice of Method

The most common methods used in research are qualitative and quantitative, and the research question to be answered determines the appropriate research method (Palgrave , 2018). According to Patton (1987), research questions in qualitative methods often starts with "*how*" or "*what*", in order for the researchers to gain a greater in-depth understanding of the topic researched. Thus, we consider a qualitative method as appropriate for studying the following:

How does agile contract templates cope with transaction hazards in ICT projects, and how does that affect when to use the contracts?

3.2 Qualitative Method

In this section we elaborate further why qualitative method is suitable for our study, supported by theoretical framework.

Halvorsen (2011) describes that qualitative research explains the non-numerical regarding the survey units, which are presented in either text or verbally. Although there are several explanations of what qualitative methods can be, Richard (2015, p. 2) describes it as "*data that are not easily reducible immediately (or, perhaps ever) to numbers*". We consider this suitable for the eventual discussion in our thesis, where the purpose is not to obtain the answer in quantifiable measures, but rather understand how the transaction hazards are affected by the contract templates in a textual form.

An additional description we find appropriate is by Askheim and Greenness (2008), that a qualitative research can be explained as a method of approaching reality, where the outcome is the production of descriptive statements or observations of the human behavior. Thus, we find the method appropriate as the research question requires in-depth knowledge of the usage of agile contract templates.

Graebner, Martin and Roundy (2012) refers to qualitative methods as a well-suited approach if the research desires to develop theory. In the process of reviewing literature for this study, we found limited articles about TCE in regard to the Agile method, and further how contract templates enable the method while coping with transaction hazards. Thus, we consider the topic of this study to be relatively new and therefore argue that there is a gap in the existing literature.

By the theory presented about qualitative research method, we believe that this method best ensures that the research question will be answered, and that the study will shed light on the importance of how agile contract templates cope with transaction hazards.

3.3 Data Sources

This master thesis contains two types of data sources: two agile contract templates and five interviews, where both sources will contribute to the analysis and discussion with equal weighting.

The contract templates are used to provide us as researcher better insight and knowledge of the requirements, processes and elements that constitutes the agile contracts. Whereas the interviews will provide personal statements, examples and insights to the experiences using the agile contract templates. As we analyze the two sources simultaneously, this allows us to draw connections and produce findings to answer the research question.

3.3.1 Data source 1: Contract templates

Two different institutions in Norway have developed standardized agile contract templates for ICT-deliveries:

- Direktoratet for Forvaltning og IKT (Difi)
  (Directorate for Administration and ICT)
- Den Norske Dataforeningen (DND)
  (The Norwegian Computer Society)

IKT Norge (ICT Norway) also provide standardized contract templates for ICT deliveries, however per date they do not hold an agile contract template, hence they are not included in this thesis.

All names will be used in their original form, in Norwegian.

3.3.1.1 Contract template 1: Difi's SSA-S

Difi is the Agency of Public Management and eGovernment, overseen by the Ministry of Local Government and Modernization. Their aim is to strengthen the Government's work in renewing the Norwegian public sector and improve its organization and efficiency (Difi, 2018).

Difi's contract templates are called "Statens Standardavtaler" (The Government's Standard Agreements), hereby referred to as SSA, and are made with input from both the customer- and supplier-side (Difi, 2018). Of nine available SSAs, there is only one agile contract template, "Statens Standardavtale – Smidigavtalen", which is elaborated below.

Statens Standardavtale - Smidigavtalen (SSA-S)

SSA-S is the Agile Software Development Agreement. This is meant to cover the customer who is in need for an IT-system where it wishes to specify in detail together with the supplier using the Agile method (Difi, 2018).

The creator of SSA-S is Mari Vestre at Difi, which later in the chapter is presented as one of the interview objects.

The structure of the SSA-S contract is the following:

1. Contract document
2. General contractual provisions
3. Appendices

The data source is retrieved from Difi's webpage for public procurement.

The contract template is accessible by everyone and is free of charge.

3.3.1.2 Contract template 2: DND's PS2000 Agile

In 1997, SINTEF and NTNU (Norwegian University of Science and Technology) formed a research program called "Prosjektstyring" (project management) where public and private players established a group who documented "best practice" in several big-sized IT projects from 1997-2000.

The research program was led by PROMIS, and the research resulted in the contract template PS2000 Standard (Prosjektstyring2000).

Den Norske Dataforeningen (DND) overtook the responsibility for further development and maintenance of the contract standard after the research program ended, along with a reference group consisting of customers and suppliers (Den Norske Dataforeningen, 2018). DND now has a total of seven contract standards, where two are based on the Agile method: PS2000 Agile and PS2000 SOL.

This master thesis will solely focus on PS2000 Agile, as PS2000 SOL is suited for larger scaled projects (100 million NOK class and above), which are not as common in Norway and not comparable in the same extent to SSA-S.

PS2000 Agile is elaborated below.

PS2000 Agile

PS2000 Agile origins from the need of a more agile method compared to the traditional waterfall process in DND's PS2000 Standard. The agile contract template is therefore similar to the earlier standard contract, but with implemented agile method in conditions and processes.

Jørgen Petersen is one of the creators of PS2000 Agile. He is being presented as one of the interview objects later in this chapter.

The contract template consists of the following parts:

1. Contract document
2. General contract provisions
3. Specific conditions (appendices)
4. Guidance

The data source can be found at Dataforeningen's homepage. The contract template is priced from 4900-6600 NOK, depending on whether the buyer is a member of Dataforeningen. Jørgen Petersen provided us with the template for this study, free of charge.

3.3.2 Data Source 2: Qualitative Interview

We consider interviews as the most suitable qualitative method for this thesis based on the notation from Johannessen et al. (2011, p. 145) that "*social phenomena are complex, and the qualitative interviews makes it possible to enlighten the complexity and nuances**". As the topic is complex, we argue that a flexible method providing insight and understanding by the interview objects' statements is appropriate.

Whereas a structured survey can limit the information flow, an interview permits the informants with greater freedom to express themselves and share experiences and perceptions. It is therefore important that the interviews are considered as a dialogue, rather than questions to be answered, to better ensure a purposeful flow of information.

3.3.2.1 Method of Interviews: In-depth & Semi-structured Interviews

We choose to use in-depth interviews to enable a relatively free conversation in order to extract as much information as possible from the informants. This form enables us to receive more extensive information, such as how the contract templates were developed, how they in practice are used in the market today, and their respective challenges and benefits.

A qualitative research interview is a conversation with both structure and purpose, where the structure is linked to the roles between the participants in the interview (Kvale & Brinkmann, 2009). In the interviews, we as researchers function as the interviewers, and the key persons involved in the cases will be the informants. Johannessen et al. (2011) separates conducting interviews by three forms: unstructured interview, semi structured (or partial structured interview), and structured interview. After considering the benefits of the three forms, our decision is to use the most widespread form: semi structured (partial structured) interviews. This interview form aims to merge the best aspects from both unstructured and structured forms, while at the same time minimize the risks (Myers, 2009).

A semi structured interview is based upon using an interview guide developed from the research question, meaning a list consisting of the themes and general questions to be reviewed in the interview (Johannessen, Christoffersen, & Tufte, 2011). Meyers (2009) argue that there should be no strict adherence to questions, and new questions can emerge during the conversation. We expect the latter to most likely occur in the interviews, where we consider new questions and its responses to be beneficial to shed light on undiscovered aspects of the research question. Hence, we take use of an interview guide as a framework but not necessarily strictly follow its particular order.

The interview guide is elaborated in sub-section 3.4.3.3 as this is affected by the sample of informants which is elaborated below.

3.3.2.2 Sampling of Informants

Sampling is an important part of the qualitative method, and in this section, we explain the choice of informants and present them later in Table 1.

We need to obtain a sample with knowledge that enables us to best possible answer the research question, thus we choose a non-random sampling: strategic selection. This method fits with the aim of this study, to develop theory and get a holistic understanding of the elements and context of the phenomena (Grønmo, 2010).

Due to the need of in-depth information regarding the contract templates and their content, we find a need for interviewing persons that have been involved in creating the contracts, which are persons at Difi and PROMIS.
Further, we also see the need for the customers' viewpoint, as they most often choose which contract template to be used. The customers' viewpoint is represented by Statens Vegvesen and Sykehuspartner. Lastly, we need information from the supplier-side to have a full circle of involved parties'

viewpoints. The supplier side is represented by Sopra Steria, because they have experience with the agile contract templates in delivering ICT projects.

We contacted all informants except from Sykehuspartner by own initiative and e-mail. The informants from Sykehuspartner was recommended by the supervisor of this thesis, Jon Bingen Sande, due to their knowledge regarding the theory and customer perspective, where Sande initiated contact on our behalf.
Some of the informants we reached out to wanted to bring an additional person from their respective company in the interview, as he or she also had extensive knowledge of the theme. This was permitted as it was considered to bring additional information.

An overview of the interview objects, their company, background and role are given in Table 1 below:

Interview object	**Company**	**Name**	**Background**	**Role of usage**
1	*Sykehuspartner*	Roar Jakobsen & Oddgeir Hvidsten	*Both* : "Sr Advisor Innovation Management" at Sykehuspartner. *Jakobsen:PhD* Dissertation in Public Sector Service Contracting with focus on TCE.	Buyer/Customer
2	*PROMIS*	Jørgen Petersen	"Managing Director" at PROMIS. Member of Dataforeningen's professional group for IT-contracts.	Leader of the creation of PS2000 templates.
3	*Difi*	Mari Vestre	"Project Manager Innovative ICT Procurement".	The creator of SSA-S
4	*Statens Vegvesen*	Ole Henrik Lidi & Svein Hauge	*Lidi* : "Sr. Advisor Procurement & Framework Agreements" at SVV. *Hauge:* "Department Leader" at SVV. Member of the board of IT-contracts for Dataforeningen.	Buyer/Customer
5	*Sopra Steria*	Nils-Petter Kristiansen & Benedicte Bjørbak	*Kristiansen* : Project Director AM/AD. *Bjørbak:* Director Applications Scandinavia	Supplier

Table 1: Informants

3.3.2.3 Interview guide

Due to different perspectives from the informants, we operate with two interview guides. The reason being that interviewing the creators of the contract templates are meant to provide more insight to the actual contracts, thus specific questions regarding these, whereas the interviews with customers and suppliers are meant to provide insights to actual use, thus specific questions regarding this.

Askheim & Grenness (2008, p. 123) argues that "*A good interview guide moves from the general to the special**"*.* Thus, the interview guide is structured with the above notation in mind, starting with general topics regarding contracts and customer/supplier relationships, before we aim to steer the interview towards the implications and impact of the different contract templates. Throughout the interviews we encourage the informants to share own experiences and examples. Additionally, we ask explicitly about the three transaction hazards in relation to the contracts.

For both interviews guides we follow the structure of Johannesen et al. (2011, p. 150): *introductions, fact questions, introductory questions, transition questions, key questions, termination*:

1) **Part 1**: We as researchers and the informants introduce ourselves, and thereby we present the research question and the aim of the thesis.
2) **Part 2**: Introductory and fact questions are merged: we start the interview with questions regarding the contract templates.
3) **Part 3:** Key questions are mostly based on our research question, the linked sub-questions, and relevant questions to enhance example and information sharing. The majority of the time is spent in this part.
4) **Part 4:** To close the interview we summate the key points from the informant, before asking if he or she want to add something. This is done to ensure that the information they see as most important, regardless of our questions, is included. Additionally, we will ask whether there is something in the contract template the informant finds missing as of date.

The full interview guides can be found in Appendix 2.

3.3.2.4 The Interviews

All informants were contacted regarding participation by e-mail. We sent a standard e-mail requesting their participation, where some lines were adjusted as to how we had heard of them. The e-mail is provided in appendix 3.

The interviews were conducted on separate dates. We started with interviewing PROMIS and Difi, in order to get deeper knowledge and insights of the different contract templates. Thereafter we continued with the customers, and at last the supplier. All informants were given the choice of when and where the interviews could be held. By having the interviews on different dates, we were able to draw from experiences and new knowledge from the previous interviews to the following, thus evolving the quality of the interviews as we moved along.

All informants were informed that we anticipated the interview to last approximately 1-1,5 hour, which was stated in the e-mail request for participation. This is aligned with Ryen's (2002) guidance, arguing that an interview should not surpass 1-2 hours. However, some of the interviews lasted longer than the estimated hour, due to the informants' extensive information sharing.

All interviews were held at the informants' respective work sites, suggested by the informants themselves. This was done due to practicality and comfortability of the interview objects as we wanted the interview to be a relatively free conversation. The latter is showed to be best achieved with a safe and familiar place (King & Horrocks, 2010).

Qualitative research often needs full record of all interviews conducted to enable a comprehensive data analysis at a later stage (King & Horrocks, 2010; Anderson, 2010). We used our own iPhone to record all interviews. All informants were asked in advance of the interview if they agreed to this method, where all accepted without hesitation.

Both of us as researchers were present at all interviews. This ensured a mutual understanding of the informants, their statements and examples. We also believe that this substantiated the effect of the conversation, which enhanced information sharing. Prior to the interview, separate roles were agreed upon, where one of us would function as the main-interviewer, also known as the moderator, and the other as a controller. In the interview, the moderator steered the conversation and asked most of the questions and explained if the informants were unfamiliar with some of the concepts. This also eased the situation for the informants as he or she would have one person to focus on. The controller was responsible of ensuring that the interview guides was used and was able to ask follow-up questions to ensure a greater understanding of the statements and examples.

By the measures taken with interview guides, audio-recording and our different roles during the interview, we ensured a sound flow of information from the informant that highly contributed to our analysis.

3.4 Qualitative Data Analysis

This section presents how we analyze the collected data from the contract templates and interviews. Data analysis is defined by Askheim and Grenness (2008, p. 142) as *"systematization of data in order for eventual patterns and structures to emerge*"*

As our method consists of two types of data sources it will accordingly impact the analysis due to structure and form. We have chosen to follow the approach of Askheim and Grenness (2008) to analyze our data, which is inspired by the well-known logic of Grounded Theory by Glaser and Strauss. The method allows us to structure and sort the data collected.

The data obtained from revision of the contract templates and the interviews can be categorized as "soft data". For the qualitative interviews, this is meant in terms of personal opinions, statements and examples from the informants. Neither of the data sources are able to measure statistically, thus we cannot quantify the results. However, the goal of qualitative method is rather to demonstrate attitudes and deeper insights of the phenomena than a generalizable outcome (Askheim & Grenness, 2008).

We transcribed all interviews from audio to text format within 1-2 days of the interview as important details can be lost if too much time passes (Askheim & Grenness, 2008). The transcribed files were used for extracting segments of the interview that are of interest (Auerbach & Silverstein, 2003).

Through extraction of the relevant information from each interview and review of the contract templates we were better able to spot the differences in the subjective information from the informants, their relation and context.

Analyzing the data can be viewed as a circular process, which entails the possibility that the analysis can go on "forever". To limit ourselves, we followed the rule of thumb by Askheim and Grenness (2008, p. 143): "*no more than necessary to make an adequate decision**".

There are three main operations in the circular process of qualitative data analysis, which is illustrated below in Figure 2:

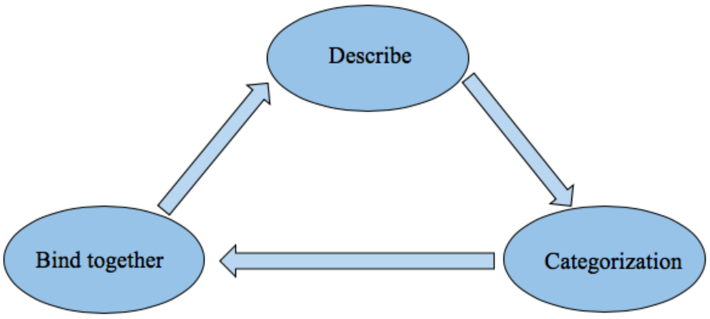


Figure 2: Operations of circular process

Step 1: Describe

The first step is to describe the phenomena as detailed and precise as possible, meaning the context must also be given (Askheim & Grenness, 2008).
This indicates that the statements from the informants must be seen in a broader context, which we have aimed for in the analysis with information and statements reproduced in their original context. Additionally, a quotation-check was sent to the informants, in order to demonstrate that no statements are withdrawn from the context it was mentioned, which is demonstrated in Appendix 4.

Whenever aspects from the contracts are used, we have strived to explain their purpose and function to demonstrate the context. An example of this is provided from our analysis where we discuss the different contractual terms, and thereafter explain the context: "*Solution Proposal – After the customer's Need Analysis, the supplier must make their comment to it and provide their Solution Proposal. This is the part where the supplier enters the co-operation with their professional opinion and solution to the customer*."

Step 2: Categorization

In the second step, the findings must be coded and structured. The coding involves sorting the text material obtained in word-codes or labels, and color the sentences with importance according to our research question and sub-questions. This later forms the basis for more substantial categorization. According to Askheim and Grenness (2008), this is probably the most used method within qualitative data analysis. During the process of both revision of the contract templates and transcription, we used this approach in order to structure the data and reduce non-relevant information. Hence, we eased the process of generating links between the data and our research question. An example of the method is shown in Appendix 5.

Step 3: Bind together

The third and final step of the circular process consist of binding together the coding and categorization from step 2. Askheim and Grenness (2008) separates by axial (focused) and selective coding. Axial coding involves taking the coded material, theory and empirics, and thereby interpret how the context and the material can be understood. Selective coding is finding the *"meaning behind the meaning*"* (Askheim & Grenness, 2008, p. 152). This indicates a more comprehensive understanding is desired, which is aligned with qualitative research. Such understanding can be obtained by collating the different data sources. By identifying patterns and correlations between the collected data material, a comprehensive understanding is more easily obtained. Thus, selective coding enables us to explain central aspects revealed by the research conducted, and thereby help answering the research question (Askheim & Grenness, 2008).

Both axial and selective coding is done, and the result of this is found in Chapter 4 and throughout the thesis.

We have strived to follow the approach from Askheim and Grenness (2008) to the best of our abilities. Given the large amount of text-based data we code and interpret, we have analyzed both contract templates and the interviews several times to best prevent losing information. Askheim and Grenness (2008) notes the strength in being more than one person multiple times. Consequently, we have worked collaboratively and double-checked each other to ensure the quality of our analysis.

3.5 Quality in Qualitative Research

In this final section of the method chapter we evaluate the quality of our research. This is categorized as validity, reliability, critical assessment of informants and ethical responsibility.

Importantly, although we are unable to produce statistical tests as in quantitative methods, we strive to show the audience that our methods are reliable and the conclusion we draw is valid (Silverman, 2005).

3.5.1 Validity

By Silverman (2005, p. 210), validity is defined as *"…another word for truth"*, and without validity the study does not obtain the required quality.
Validity in social sciences refers to whether a method is suitable for the phenomena of the study. There must be a statement of validity in the research regarding the methods used for data collection, interview and analysis (Kvale & Brinkmann, 2009; Postholm, 2005), which we provide below.

Circular process of theory, data collection and analysis

Qualitative research imposes the challenge of the validation-problem. This occurs as the researcher is captured in both theory and empiricism, where one must work with formulation of research question and interpret the result of empiric research in addition to collecting and process data (Halvorsen, 2011).

In order to face the validation-problem we have worked with the relevant parts of the thesis (theoretical frame of reference, method and analysis) collectively, meaning that no part was completed prior to the next, but rather worked in a circular process in regard to each other. This is done to ensure that the parts are coherent and clearly structured to increase the validity.

The Interview Objects

We have strived to solely interview people with extensive expertise regarding the topic of agile contract templates. Our research question is not common knowledge for the general population, thus we used strategic selection to acquire informants with relevant knowledge. The interview objects are found trustworthy with extensive experience of ICT projects and have all used both traditional Waterfall models and agile methods. Additionally, when speaking with the different informants, we found that most of them were acquainted with one another, and also recommended each other for interviews. We therefore find that the agile environment of those considered experts is quite small, and that the informants that participated in this study classifies as an appropriate selection.

Statement of Validity

Throughout the current chapter, we have strived to explain why and how our choice of qualitative research was done with supporting theory of how the research method fits our research topic. Thus, we have aimed to demonstrate both relevance and credibility in our methods. We have included an explicit description of our data sources and their relevance, followed by a detailed clarification on how interviews were conducted and lastly the method used for analyzing our results. This is done to ensure the validity and give the audience a sound insight to our procedures.

It is important to note that any research have weaknesses, factors, perspectives or parts that are not included, or factors that influences the study, hence affects the validity (and reliability) (Johannessen, Christoffersen, & Tufte, 2008). Limitations of this study are therefore presented in Chapter 5. However, based on the above notations, we argue that the validity is maintained in our study.

3.5.2 Reliability

According to Kvale and Brinkmann (2009), reliability concerns whether the research can be reproduced by another researcher at a different time, and still obtain the same results.

However, one must separate between quantitative and qualitative studies and their respective reliability. Whereas in quantitative studies it is possible to test and measure reliability by standardized method, this is not applicable in the same degree for qualitative studies (Grønmo, 2010). This is due to qualitative research being less structured where the data collection is not separated as an own phase, but function as an ongoing element alongside of the analysis and discussion.

Ongoing Development

The above notation is true for this study, as the two data sources are both collected and developed as the study has evolved due to new information occurring from the interview objects. Thus, reliability for qualitative studies must be based on systematic discussion of the different elements in the research method and data collection (Grønmo, 2010). We have therefore explained and extensively described the elements of our method that collectively forms the thesis throughout this chapter. Through presentation of our method we have tried to demonstrate that the findings are not drawn from our own subjectivity but are rather actual facts provided by the two contract templates and our informants' expertise. Additionally, by detailed explanation of our process prior, during, and after the interviews, we strive to show that our data are collected systematically and in compliance to established procedures, hence that the assessment of reliability strengthens the findings in our study.

Stability

According to Grønmo (2010), reliability can among other things be explained by stability, which is the degree of consistency between the data collected at different times. To evaluate stability, he argues that the researcher can do a critical assessment of the same data on different times. Following Grønmo's guidance, we reviewed the data sources and the findings drawn from the sources on multiple occasions. As previously mentioned, qualitative method contains an ongoing analysis phase, where input from each interview gave us more insight to the contract templates, which made us discover other aspects from previously held interviews. Therefore, we have gone back and revised our data, thus also findings, several times. This is done to ensure stability by clarifying our descriptions and eliminate possible sources of error.

Two researchers

The strength in more than one researcher has been mentioned multiple times by various literature used for the current chapter. Both us as authors of this thesis have participated equally in all parts of this study. The analysis and findings are first revised individually and thereby collectively in order to ensure that our findings are as objective as possible, despite the acknowledgement that qualitative methods will always be somewhat influenced by the researchers' subjectivity (Grønmo, 2010).

Statement of Reliability

We have strived to ensure transparency with our method, and the previous sections in this chapter have provided a detailed explanation of how we have collected and analyzed our data sources. The interview guides and illustration of the analysis process are included in the appendices in order for the reader to further understand our method. Additionally, by demonstrating our method and process in entirety, we have tried to ensure that the study can be verified, and thus attempted to ensure reliability.

3.5.3 Critical Assessment of Informants

As one of the main points behind choosing a qualitative method is allowing the informants to speak freely regarding their expertise and knowledge, we must remember that the provided information is related to their background and view-points. Hence, the data gathered from informants must be considered somewhat subjective.

An example of this is Lidi at Sykehuspartner who has extensive knowledge of TCE due to his academic background. Because of this, his statements could be inflicted a higher relative importance as it is well-grounded in theory. When Lidi answers to our questions, there is reason to believe that his reflections of what he considers "most important" or "biggest issues" are influenced mostly of his understanding of TCE, rather than observations from agile ICT projects in practice.

Another example that must be considered is that the creators of the contract templates naturally will consider their respective templates as the better option, and therefore find problems with the other.

Hence, we must assume that all our informants to some extent are subjective in their statements due to their role as either creators, buyers or suppliers. However, the analysis is based on a consideration of the complete data set of agile contract templates and the interview objects, and therefore always seen in context and relation to each other when included in the analysis.

3.5.4 Ethical and Legal Responsibility

One can argue that the use of qualitative methods implies an even greater responsibility for ethical assessment, as our process includes in-depth interviews, thus forms a relation between us as researchers and the informants. In this final section of this chapter we explain how we assess the ethical responsibility.

We have throughout the process of this study strived to comply with the guidelines of "The Norwegian National Committees for Research Ethics" (2014). In summary, these guidelines are covered by 10 principles that we have followed, where we below explain the most relevant to our study.

The fourth principle is "Voluntary informed consent". In the initial e-mail sent to the informants they received a presentation of us as two master students from BI Oslo, how we had heard of them (for example by recommendation), the purpose of our study and then a request for participation. All informants agreed to participate and seemed very interested to contribute. According to the guidance by Johannessen et al. (2011), the informants can at all time withdraw the consent to participate, which was stated to the interview objects pre-interview. Then the informants were asked for consent to audio-record the interview, where all approved immediately.

The fifth principle is “Confidentiality”, where we as researchers must avoid any use of information that might inflict damage on the individuals. We asked if anonymity was desired prior to audio-recording and starting the interview, where all participants answered no. However, if any sensitive information would have been brought up despite the approval, we would have eliminated this from the thesis in order to comply with the principle.

The sixth principle is “Impartiality”, which is avoidance of confusing roles and relationships. By presenting ourselves as master students in the initial contact with the informants, we established directions for the relation, and reduced expectations of returning benefits, such as for example payment for participation.

In our opinion, we are also compliant with the ESOMAR’s Code of Practice (European Society Opinion and Marketing Research) guidelines, which includes: volunteerism, anonymity, notification of observation or recordings, later use of audio- and video recording (Askheim & Grenness, 2008).

4. Analysis of the Agile Contract Templates

In this chapter of the master thesis we analyze our data in order to answer the research question: *How does agile contract templates cope with transaction hazards in ICT projects, and how does that affect when to use the contracts?*

To answer this question we have, through our review of the contract templates and interviews, derived seven themes that affect the extent of agility and the transaction hazards. An overview of these themes within the contract templates are provided as a table in section 4.2. Furthermore, a more thorough analysis and comparison of the contracts concerning these subjects and their effect on the transaction hazards are given.

4.1 Explanation of Terms

Both agile method and the contract standards follow a specific terminology which we use in our analysis. To ensure that the reader of this study fully understands the meaning of these terms, they are explained in Table 2.

Term	**Explanation**
User Stories	Agile method tool meant to capture smaller descriptions of software features. Told from the perspective of the end-user.
Product Backlog	Prioritized feature list, meaning what feature is to be developed before the next one, and categorization of how important that feature is. A to C. Agile method-tool of replacing traditional requirement specification.
Sprint	An iteration. Set period of time where specific work has to be completed.
Non-functional demands	Technological demand; performance, capacity, response time, security, etc.
Fixed Price	Non-negotiable set price for product or service.
Target price	Estimated total sum for completing the project.
Team price	Total hourly price rate for the team members.
Project Triangle	Triangular model of the constraints of project management. The quality is constrained by time, cost, and scope.

Table 2: Explanation of terms

4.2 Analysis of the Relevant themes in the Contract Templates

Table 3 presents an overview of the seven themes we identified as relevant by affecting agile development projects and how the contracts templates cope with the three transaction hazards. We sort by contract template type, similarities and differences:

	SSA-S and PS2000 Agile	SSA-S	PS2000 Agile
	Similarities	**Differences**	
Need Analysis	User Stories & Product Backlog	None	
Solution Proposal and Scope	Supplier must answer to customer's Need Analysis.	Open format.	Strict format of two phases: Preliminary solution description and Solution Description Phase
		Excpects change.	Expects comittement.
Risk Analysis	None	Risk management: Not specified. Customer bears most risk.	Risk management: Risk Matrix. Customer can split risk, can delegate to supplier.
Price Model	None	Team price	Fixed price +Target price +Cost additions
Project Group	Product Owner from customer.	Low degree of spesifications.	High degree of spesifications.
Delivery Plan & Competency	Delivery plan based on Iterative processes as management tool. Led by Product Owner. High degree of customer involvement and competency required.	Change order with low degree of spesifications.	Change order with high degree of spesification.
Termination	Early exit & ongoing exit option for the customer.	Low cost of termination.	High cost of termination.

Table 3: Contract overview

In the sub-sections below, we present and analyze the seven themes in detail. Each section is structured as follows:

- An excerpt from Table 3 of the relevant theme is presented
- Themes that are similar within both contract templates are presented in a summarized section.
- Themes that are affected differently by the contract templates are divided into sub-sections.
- Each theme is summarized by the most important similarities and/or differences in relation to the three transaction hazards.

4.2.1 Need Analysis

	SSA-S and PS2000 Agile	**SSA-S**	**PS2000 Agile**
	Similarities	**Differences**	
Need Analysis	User Stories & Product Backlog	None	

Table 4: Need Analysis

Similarities among the contract templates

Both contract templates have similar principles with regard to the Need Analysis. They do not differ in any relevant way in their relation to this theme, and so they are analyzed simultaneously in this section.

In the Need Analysis the customer specifies their functional and non-functional demands for the software to be developed by a supplier. It is crucial that this is done in a manner that allows the supplier to easily respond with their Solution Proposal that describes how they will develop the customized software. Both contract templates promote the use of User Stories and Product Backlog with priority categorization. Hence, to give guidance of order for the software development and solution it is created a prioritization of what to develop first.

Summary of Need Analysis and its effect on transaction hazards:

We conclude that the Need Analysis in both agile contract templates are similar with their focus on User Stories and Product Backlog. The Need Analysis reduces uncertainty as it structures the premises for the software delivery.

4.2.2 Solution Proposal and Scope

	SSA-S and PS2000 Agile	**SSA-S**	**PS2000 Agile**
	Similarities	**Differences**	
Solution Proposal and Scope	Supplier must answer to customer's Need Analysis.	Open format.	Strict format of two phases: Preliminary solution description and Solution Description Phase
		Excpects change.	Expects comittement.

Table 5: Solution Proposal and Scope

Similarities among the contract templates

After the customer's Need Analysis, the supplier must respond by providing their Solution Proposal, which entails entering the co-operation with its professional opinion and solution to the customer. Furthermore, both parties jointly consider the scope of the software development and its relevant estimates e.g., cost, time line, resource allocation, in order to gain an understanding of the project's complexity.

It is important that the supplier evaluates its own ability to deliver on the scope, and that it provides accurate estimates of the project. The possibility of erroneous estimates, which leads to a high degree of uncertainty and risk, is something that happens frequently, according to Statens Vegvesen. Hauge notes that suppliers often provide unrealistic Solution Proposals while the customer lacks the competence to recognize this, thus leading to an underestimation of the project's complexity. The agile contract templates, however, provide a different approach to address estimates in their Solution Proposal and scoping phase.

Differences among the contract templates

The contract templates differ in the solution description and scoping phase of the extent they have implemented the Agile method. SSA-S is promoting an open-ended and flexible scope, whereas PS2000 Agile tries to commit the parties to a more rigid scope of a detailed solution description.

SSA-S:

Under this template, the supplier is responsible for guiding the customer through the different solution alternatives that will fulfill their non-functional demands. Hence, arriving at the satisfactory non-functional demands is mainly the supplier's responsibility. If the supplier discovers that the solution chosen by the customer will harm their desired demands, the supplier must notify the customer through written communication.

Vestre from Difi notes that the solution scope in SSA-S is meant to be an open-ended solution where the customer is not locked to detailed and rigid specifications for the solution they want; "*The solution will therefore mature as the project moves along, and thereby it is supposed to create the solution the customer actually need, rather than what it believes it needs*". Vestre argues that this is aligned with the Agile method, which corresponds with Statens Vegvesen's experience where Hauge notes: *"SSA-S enables the use of the Agile method and is a lot less binding."* Moreover, according to Vestre, this is opposite to PS2000 Agile, which she considers more aligned with the Waterfall model.

PS2000 Agile:

Our informants consider the Solution Proposal and the rigid scope of PS2000 Agile as a comprehensive phase, thus it does not correspond with the Agile method. Below we present the different elements of the PS2000 Agile's Solution Proposal.

With this template, the supplier's response to the Need Analysis is given through a preliminary Solution Proposal, followed by a sub-section concerning deviations and assumptions found from the customer's Need Analysis. Petersen from PROMIS states that *"[The preliminary Solution Proposal] is the first after contract signing where the parties are to co-work on the solution and acknowledge their shared responsibility"*.

After the preliminary Solution Proposal, the supplier must, within the established time-limit, develop the Solution Description Phase (SDP). This is carried out jointly by both parties, with the main goal of deriving at a common understanding of the demands and the solution of the delivery. Therefore, the most important component of the SDP is an estimated and prioritized Product Backlog. Petersen emphasizes that "*It is crucial that both parties truly make all necessary clarifications, in order for the supplier to decide if they indeed are able to deliver on their scope".*

After that, the SDP must be approved by the customer before it is considered as the updated (final) Solution Proposal, which forms the basis of the software development. Petersen explains that *"The supplier has committed to the delivery in the solution description, and if not terminated after the first phase, the delivery must be completed".* Thus, if the supplier at this point verify its ability to deliver on the scope, it is committed to do so. Petersen pointed out that this commitment is a key difference between PS2000 Agile and the SSA-S, because in the latter, the supplier's commitment is more diffuse as it is not aware of the whole scope.

Sopra Steria is very positive to the estimation model in PS2000 Agile, which is a main part of the SDP, and provides the opportunity to break down the project's budget. However, the supplier recognize that it is difficult to have a complete understanding of the project in the initial phase, meaning that there can be changes to the estimation model in later stages during the software development. This is typically expansion of scope due to the software needed is found to be more complex than first estimated, change of prioritization in the Product Backlog and so on, and therefore Sopra Steria experience that the estimation model is often updated in later phases.

Summary of Solution Proposal and Scope, and its effect on transaction hazards:

The Solution Proposal phase raises similar response-requirements to the Need Analysis for both contract templates. Here also, it is crucial that the supplier evaluates its ability to deliver the project, as wrong estimates creates uncertainty regarding the project's evolvement. Both we and the informants recognize that the entirety of the scope cannot be known in the initial phase of an agile project, and therefore, regardless of contract template, the method causes some uncertainty around time and cost estimates.

While SSA-S offers an open-ended solution scope, PS2000 Agile is more rigid with its two phases that combines to become the final solution description that the supplier is committed to deliver on. The open-ended solution scope of SSA-S can, as Petersen argued, be considered more diffuse and therefore also leading to more uncertainty into the project's development and final cost. This aligns with the Agile method.

We find that because both parties agree on an end-result, the PS2000 Agile is able to cope with uncertainty and performance ambiguity. By facilitating a shared understanding of the project, PS2000 Agile eases accurately estimations of the project's time and cost. Moreover, as it creates a rigid plan for what the supplier will develop, it avoids performance ambiguity caused by misunderstandings.

4.2.3 Risk Assessment

	SSA-S and PS2000 Agile	**SSA-S**	**PS2000 Agile**
	Similarities	**Differences**	
Risk Analysis	None	Risk management: Not specified. Customer bears most risk.	Risk management: Risk Matrix. Customer can split risk, can delegate to supplier.

Table 6: Risk Analysis

None similarities: Two different thoughts of risk management

There are no similarities in how the agile contract templates deal with risk management. Through our interviews, we find that risk management in agile projects is mainly influenced by two different principles of risk assessment. Thus, before elaborating on how the agile contract templates assess risk, these two principles are described below.

The interview objects differ in their opinion of how to consider risk at the initial phase of a software development project. On one hand, some argue that early risk assessment and delegation of responsibility is not feasible because some risks will not become apparent until after the project starts. These sentiments thus align with the Agile method, which prescribes both parties to deal with risk when it occurs by reporting it to the project group so that corrective measures can be implemented. On the other hand, some informants argued that risk evaluation and delegation of responsibility should occur before signing the agreement.
Among others, Statens Vegvesen ascribes to this view, as they describe their risk assessment as an evaluation of their internal competency and ability to affect risk factors, thereby bearing the risk themselves. Another option is to sign an insurance which makes the supplier responsible for the risk. Diverging views on how to consider risks should therefore influence choice in contract template as the latter also addresses risks differently.

Differences among the contract templates

We find that SSA-S and PS2000 Agile assess risk differently in their frameworks, where the main difference is that PS2000 has a Risk Matrix.

SSA-S:

SSA-S does not have an annex or section specifically devoted to risk analysis. This corresponds with the Agile method where coping with risk on an ongoing basis is central.

Vestre from Difi explains that even though the Agile method does not have rigid structures for risk management, some initial agreement is necessary. She notes that a risk element cannot be shared, and therefore distribution of risk has to be defined as accurately as possible. However, as noted, SSA-S does not provide any guidance on this in the appendices. Hauge at Statens Vegvesen therefore argue that SSA-S can become redundant as it does not provide them as a customer with any contractual benefits in terms of coping with risk.

Difi and Statens Vegvesen noted that in SSA-S, the risk is mainly held by the customer. This is because they are responsible to facilitate the supplier’s delivery. If this is not done correctly, and the project halts because of it, the customer is punished through payments to the supplier as the latter must wait to continue work. As Vestre describes: “*The customer bears the risk of making sure the supplier’s staff has work to do as it is dependent on the customer to make decisions which gives the staff directions for ongoing work”.*

PS2000 Agile:

PS2000 Agile aims to cope with risk through The Risk Matrix. This matrix is to be assessed prior to contract signing and updated after the Solution Proposal. An excerpt of the Risk Matrix that outlines its form and function can be found in Appendix 6.

The Risk Matrix functions as a comprehensive check-list that identifies and describes the risk elements, including the parties’ assessments of them. It also links the different risk factors with specific actions and persons that are responsible for delegating these actions to the customer or supplier, something which Peterson at PROMIS describes as very important. This delegation forms the foundation for the risk-addition that can be included in the contract price for those elements the supplier is responsible for.

According to Petersen, the matrix constitutes a central aspect of the contract template as it enables both parties to put a price on perceived risks. If the customer identifies the risk as high, it can be added as an additional cost into the contract’s Target price, which then acts as an insurance where the supplier must cope with occurring risk scenarios. However, the customer can also choose to bear the full responsibility of risk, meaning the additional cost is avoided.

Risk Matrix in Practice

Sopra Steria’s experience is that the Risk Matrix is not used in the extensive way it was thought out to, and that it is contradictive to the Agile method. They find it difficult to accurately assess risk in the initial process of the development project as there will be changes to the scope and project during the later software development phase. Therefore, they find it troublesome to delegate risk and calculate its potential price into the project's estimate. However, they still perceive risk management an integral part of all types of projects, and that a good risk assessment provides the foundation for further collaboration.

Bjørbak from Sopra Steria argued that when a problem occur details of it will follow, which in turn give more insight of how to cope with it. This means that the risk is not the same as initially perceived in the Risk Matrix, but rather a development of it that later has become a problem. Thus, the important factor is to establish a management regime with a standardized way of reporting risk factors, and when the problem occurs, both parties must decide on the ownership of the risk, actions to be taken and a time-limit for their implementation.
Hence, use of the Risk Matrix can to some extent be similar to how Difi uses SSA-S for risk management, where they manage problems as they arise.

Nonetheless, Petersen claims that even though many argue that it is impossible to calculate risk in this manner, the Risk Matrix has proven to be successful. He also emphasizes that by forcing the parties to consider and evaluate potential risk factors, they become aware of future risk scenarios, thereby allowing them to better cope with problems if they occur.

Summary of Risk Assessment and its effect on transaction hazards:
The agile contract templates differ in how they deal with risk management.
SSA-S does not specify how to deal with risk, something which aligns with the Agile method. Yet, we find that this increases uncertainty. If problems arise during a project with few stipulations regarding risk management, both parties are less liable to deal with them, and thus there is a higher potential of unforeseen costs related to recuperation.

Although the Risk Matrix is not always used diligently, we find that by creating awareness of potential risk scenarios, both parties have the possibility to be pro-active in coping with risk elements before they become a problem later in the project. Hence, decreasing the uncertainty for consequences, e.g. costs related to occurring problems. If a risk element is not manageable when assessed in the Risk Matrix, the template provides the opportunity for the customer to delegate the responsibility to the supplier by paying them a cost addition included in the target price, i.e., an insurance coping with uncertainty related to risk for the customer.

4.2.4 Price Model

	SSA-S and PS2000 Agile	**SSA-S**	**PS2000 Agile**
	Similarities	**Differences**	
Price Model	None	Team price	Fixed price +Target price +Cost additions

Table 7: Price model

None Similarities: Different price models

Both contract templates allow different price models based on the project. The informants revealed that the most commonly used price models in the two agile contract templates are hourly-based rates, fixed price and target price.

Differences among the contract templates

Whereas PS2000 agile has a price model consisting of fixed price, target price and cost additions, SSA-S promotes the use of hourly team price.

SSA-S:

SSA-S provides more flexibility in its contract template in regard to price models. Here, the customer can choose the model they see fit, e.g., hourly team price, fixed price, target price or running hours. Yet, the agreement encourages a team price based on the hourly rate of the participants in the team. As team collaboration utilizing different experts in the development is considered important in agile development projects, we find this to align with the Agile method.

Vestre explained that the creator group of SSA-S considered target price in their agile contract template but did not find it suitable in an agile project as the point is to fulfill a need during the development, rather than having a pre-specified target. Also, Vestre chose to avoid the fixed price model because it requires the supplier to work for free if a project is more complex or needs more work-hours than initially anticipated. Vestre stated that this is due to the 'Project Triangle', where quality is prioritized over time and price: *"If the agreed quality is not provided, [the supplier] must work for free".* Hence, according to Vestre, professional developers will never choose fixed price on software development projects.

Nonetheless, Sopra Steria experience that in practice, those in the public sectors who utilize SSA-S try to implement a fixed price model into the contract. According to Sopra Steria, this is due to the public sector's requirements of detailed budget estimations regarding procurement, thus making it difficult to undertake projects created solely on hourly based price without a clear scope.

PS2000 Agile:

PS2000 Agile's price model comprise several price elements that jointly form the total price estimate. These elements are fixed price, target price and cost additions, where the first two are found as contradictive to the Agile method because they are pre-decided and limits the total cost.

Firstly, the fixed price contains elements that are independent of scope for the supplier's work hours, e.g., software programs and hardware included in the project's delivery. Secondly, the target price refers to the cost associated with the part of the delivery that is based on hourly work, e.g., the solution scope. This is the supplier's most realistic estimate of the number of hours the delivery and each Sprint will consist of. Additionally, the target price can also consist of the cost found in the Risk Matrix, where a price has been linked to risk elements delegated as the supplier's responsibility during the project.

Finally, the price model also includes cost additions that are divided in two, where the first is an addition for aid related to the acceptance test, and the second is an addition for the work related to fulfill the commitments in the contract period.

Example of the price model in practice

Lidi from Statens Vegvesen stated that projects using fixed price are approximately 30 per cent more expensive than those with hourly-based rate, due to the risk of the project being priced and incorporated into the price model because the supplier holds the full responsibility for the software delivery. However, as noted in the section on risk assessment, Hauge argued that if Statens Vegvesen feel safe regarding the delivery, they find it unnecessary to pay the extra cost of including insurance in the target price.

Statens Vegvesen has negative experiences with fixed price contracts where the supplier wins the contract based on a price model with a low profit margin for the supplier that makes it difficult for them to earn good enough profit on the contract. Hauge noted that this is a rather common behavior among suppliers to win contracts, where they later are trying to expand the scope with change orders of running hours so that they as a supplier makes a higher earning on the project. Vestre, and Jakobsen and Hvidsten at Sykehuspartner also mentioned this supplier-behavior as a well-known problem when using fixed price-contracts.

Example of target price in practice

Statens Vegvesen noted that the use of target price in PS2000 Agile helped them, as a buyer, to cope with an underestimation of complexity during one specific project which totaled 4 million NOK more than the initial estimation. Due to the contractual framework, Statens Vegvesen was able to split the exceeded cost with the supplier. According to Hauge, this demonstrates the usefulness of the price model of PS2000 Agile when undertaking projects with a lot of uncertainty, as both parties share the risk of additional cost when the scope is uncertain. However, this is only possible if the parties' ability to affect risk scenarios are balanced, if not, Hauge argued that the use of target price is pointless.

Hauge further added that if the project had utilized SSA-S with an hourly team price, the delivery outcome would have remained the same, but the customer would have been forced to pay the entire cost overrun.

Summary of Price Model and its effect on transaction hazards:
The two contract templates differ in their preferred choice of price model, which affect the degree of agility in the project, and in turn which party that bears the majority of the risk, thus affecting uncertainty.

From the customer's viewpoint, we consider fixed price and target price as good solutions to cope with uncertainty of the final cost. However, problems arise when the supplier works on a fixed price-project with in-adequate margin, and therefore act opportunistic to increase earnings by trying to implement more work hours based on hourly price rate in the development. This leads to uncertainty for the customer as it loses control over any plans the supplier has for change orders. This will, in turn, affect how the project evolves, and what its final cost will be.

Nonetheless, if the parties decide on a more agile development project, a price based on the team's work hours is more suitable. In this case, the customer must accept uncertainty regarding the final price of the project, as no one knows the exact hours required for the project, or the problems that might occur during the agile development. As SSA-S leans more towards the Agile method, we also find that the uncertainty of the final cost allows the value outcome to be of higher importance than budgetary transgressions.

4.2.5 Project Group

	SSA-S and PS2000 Agile	**SSA-S**	**PS2000 Agile**
	Similarities	**Differences**	
Project Group	Product Owner from customer.	Low degree of spesifications.	High degree of spesifications.

Table 8: Project group

Similarities among the contract templates

Both contract templates demand specific investments from both parties of human resources for the project group. Present in the group are both parties' appointed project leader and other key personnel, with mandates and responsibilities for their respective parties' obligations towards project delivery. Hence, the project group establish the co-operation between the parties, and leads the agile project.

The leader of the project group is the Product Owner, who is an allocated resource from the customer, with mandate to decide functional demands and needs on behalf of the customer, and who is responsible for communicating with the supplier. Thus, we find that both contract templates place a large responsibility on the Product Owner for successful project execution. Hence, we recognize that the Product Owner must have competence in both project management and technical skills in order to comply with the role's requirements and lead the development project.

Because of the agile contract templates' requirements of the Product Owner and other key personnel in the product group, we find the templates to demand high customer involvement during the project.

Differences among the contract templates

We find that SSA-S has a lower specification of how the project group should be, whereas PS2000 Agile has a high degree of specifications, where key personnel must be described in detail.

SSA-S:

SSA-S does not provide comprehensive details regarding the project group but requires that the supplier's key personnel are stated in an annex of the contract's appendices. In the part regarding project management in the contract it is specified that the supplier is responsible for having the delivery conducted with sufficient qualitative and quantitative resources and competency.

The low degree of specifications regarding the project group is considered to be consistent with the Agile method, as it stands in opposition to a strict framework of how to manage the project.

PS2000 Agile:

In PS2000 Agile, the organization and work form of the project group are stated in an own section of the contract template, with a high degree of specification, thus considered very comprehensive to administrate by the interview objects. The parties must, in an inclusive way, provide written descriptions of details regarding several parts of the work, and all staff must be specified in percentage availability, timeframe of work, function, cost and quality etc. Additionally, the parties can agree upon appointing an external third part that reports to the project group in order to safeguard the overall quality assurance.

Sopra Steria stated that the rigidness of the framework, by its clear demand of the project group and needed human resources and competency, is what makes projects successful. This supplier noted that PS2000 Agile gives clear instructions to the customer about what is anticipated of involvement and resources from their side as a buyer to a greater degree than SSA-S. Therefore, the guidelines in PS2000 Agile is considered by Sopra Steria to lay the foundation of great customer involvement.

Summary of Project Group and its effect on transaction hazards:

Both agile contract templates require allocation of human resources to the agile project but differ in terms of rigidness of requirements. SSA-S has fairly open-ended guidelines, whereas PS2000 Agile has a lot of specifications.

The Product Owner is central in both frameworks, with its leadership of the project group and its need of a certain skill-set. This specific human resource allocation is considered as asset specificity, as it is needed to lead the agile project. Also, other resources from the supplier of professional developers within expert areas can be considered being asset specificity to the project group.

Further, we find that the project group can enable a great foundation for co-operation, which creates a relational governance of the project. Therefore, we find that co-operation with competent human resources lessens uncertainty as it enables a relationship that builds mutual trust and solidarity within the software development project.

The high involvement from the customer is however sometimes not fulfilled. In Vestre's experience, the customer is more used to order production, and does not fully understand the workload of creating IT-systems. Thus, in agile projects the customer underestimates the required time and effort it has to allocate in order to create a successful result. Lack of involvement can therefore create uncertainty in the agile project and its value outcome. However, we find that PS2000 Agile copes with potential lack of customer involvement through its comprehensive demands towards the customer of its resource allocation.

4.2.6 Delivery Plan and Competency

	SSA-S and PS2000 Agile	**SSA-S**	**PS2000 Agile**
	Similarities	**Differences**	
Delivery Plan & Competency	Delivery plan based on Iterative processes as management tool. Led by Product Owner. High degree of customer involvement and competency required.	Change order with low degree of spesifications.	Change order with high degree of spesification.

Table 9: Delivery Plan & Competency

In this section we present the contract templates' take on the delivery plan and the competency required of the involved parties. Underlying the delivery plan are iterative processes and change orders, which we elaborate in sub-sections concerning similarities between the contract templates. Thereafter follows an assessment of the differences in the delivery plan and change orders between SSA-S and PS2000 Agile.

Similarities among the contract templates

The supplier holds the primary responsibility of the delivery plan, and both contract templates demand an updated and available version at all times. Thus, the templates are pushed to be used as a management tool during the project. Sopra Steria noted that changes to the scope and the project can occur during the delivery phase due to wrong estimates in the Solution Proposal.

How this is coped with depends upon the involved parties and their ability to co-operate, meaning if they have established mutual trust and solidarity in the project, which is turn enables them to cope better with changes.
However, when changes occur, it is the Product Owner’s responsibility to make timely clarifications and verifications that enables the supplier to obtain an updated delivery plan and maintain the progress of the software development.

Iterative Processes

For both contract templates, an agile project is accomplished through the delivery plan consisting of iterative processes and Sprints. The aim of this process is to develop, test and re-do until the software delivery is approved by the customer’s Product Owner. Due to the iterative processes, User Stories and Product Backlog are crucial to guide the supplier’s team on what to develop next and the time-frame for testing. In agile projects, the largest business-value for the customer must be prioritized, which in turn must be demonstrated in the Product Backlog through categorization. The templates distinguish between three categorization priorities: A (absolute), B (will be included, but can be delayed until the next Sprint), and C (desirable, but can be omitted). As prescribed by the Agile method, this prioritization can change during the project, thus the Product Backlog must be updated at all times and be transparent for both the customer and supplier.

Sopra Steria highlighted the prioritization as a way of dealing with change based on the Agile method. An example of change is when the supplier is allowed to state that they will deliver a minimum of 80% of what is planned, and where the parties accept that the remaining 20% did not make it. The latter is accepted because the percentage consists of aspects with the lowest value for the customer. Kristiansen at Sopra Steria argues that *“one should always include the most important [features], and this is the strength of the Agile method. Both contracts safeguard this”.*

Change Orders

Kristiansen noted that changes during the delivery is normal as the involved parties gets deeper into the project and gradually gains a better understanding of the project and the software to be developed, which often increases the scope. Kristiansen described this by saying: *"Research shows that after the initial analysis has taken place, the scope often increases up to 50%"*. When the project group finds that the initial estimates are wrong, the delivery plan will therefore be incorrect, thus also the execution of the project development. Hence, the parties have to find a way to solve it, which can be done through change orders.

The two contract templates allow the customer to request changes by adjusting the scope and content in the agreed delivery plan. Just as the customer, the supplier can also hand in a change request if the customer demands work or delivery that the supplier consider as outside of the delivery plan. Additionally, the supplier can request change if delayed by the customer.

When a change order does occur, the Product Owner must have the competence to verify these changes, and subsequently change the Product Backlog based on an updated categorization showing the prioritization of next development.

As noted by Hauge from Statens Vegvesen, change orders can also be used by the supplier to expand the scope when using a fixed price-contract, thus making more revenue from additional work by the developers in the project with hourly price rate on running hours. Hauge additionally stated that in such scenarios, the supplier withdraws its best consultants and replace them with less competent developers as they consider the project to bring low value for them. These types of change orders from the supplier becomes demanding and time-consuming for the Product Owner as the orders must be evaluated, and this can prolong the project

Competency

In order for the iterative processes and Product Backlog to function optimally the Product Owner must have the right skill-set of technical competencies in order to lead the project group and verify the supplier's deliveries in a timely manner that prevents stops in the software development. Below we consider what can happen in the delivery phase if there is a gap of competency and allocated time, and how some factors in the agile method can cope with this.

The informants from Statens Vegvesen and Sopra Steria experience that the iterative processes and Product Backlog are often not executed well enough by the Product Owner, which can create a stop in the project. Both Difi and Statens Vegvesen argued that a possible consequence of not having timely clarifications is that the supplier continues developing on its own to avoid a stop. Thus, if discussions regarding delivery performance later occurs, the customer often states that the delivery is not as ordered, whereas the supplier often states that the postponement caused by the customer is to blame for the erroneous delivery. Hence, conflict can arise. However, Kristiansen at Sopra Steria noted that if the supplier continues developing parts of the software while awaiting the customer's clarification, it is at the supplier's own risk.

To cope with potential waiting time and eliminate postponement during the development phase, the informants recognize that it is useful to have enough development material in the Product Backlog in order to have two Sprints running simultaneously. Hauge noted that he emphasizes this to his Product Owners to make sure they have the competency of how to run more than one Sprint.

All informants repeatedly pointed out the importance of the customer's competency and involvement during the delivery process as this reduces the risk of incorrect development or extra hourly costs without development.
The informants further mentioned that in order to secure the best possible development, the customer should match the supplier's expertise in some areas, as this enables co-working with the supplier, and helps manage the development to a greater extent.

Differences among the contract templates

Flexibility during the development is an important principle in the Agile method. The contract templates differ in how they comply with flexibility; whereas it is fairly easy to implement a change order a change order with the SSA-S, PS2000 agile demands more work to implement it. Below we consider the differences in the delivery plan and change orders.

SSA-S:

Delivery Plan

SSA-S demands both parties to jointly create a comprehensive test strategy on performance and completion of testing rounds, where the aim is to make it easier for the customer to follow the project development and be involved in the delivery plan. The customer must also pre-define the approval criteria to be used by the Product Owner for the evaluation of delivery.

Vestre at Difi noted that despite good specifications in a delivery plan, errors of the functionality and general problems during the development are common in development projects. She argued that by using SSA-S, both parties will acknowledge the lack of functionality if it occurs, and thereby re-create and solve the problem, as opposed to other contract standards where this must be handled through more comprehensive administrative work. According to Vestre, this is the benefit of using SSA-S.

Petersen at PROMIS noted another inherent challenge to SSA-S that can occur during the delivery due to lack of a specified delivery commitment towards the supplier: "*The supplier is obligated to deliver resources to work for the customer, and this is the only delivery commitment*". Petersen finds that SSA-S only commit the supplier to the work method and satisfying the customer, and not to the Solution Proposal. He believes that this can lead to difficulties in determining whether clarifications were done in order for the supplier to be able to conduct an appropriate delivery, and what was agreed upon by the parties to achieve satisfactory delivery.

Change Orders

The guidance in SSA-S states that in agile development, changes of the software should be handled similar as other demands and needs, thus there is no need for a specific framework of change order management. This is aligned with principle 2 in the Agile Manifesto which is one of the most important ones according to Vestre*: "Welcome changing requirements, even late in development"* (Beck, et al., 2001).

However, the agile contract template states that the customer is responsible of specifying the demands of the delivery, meaning the potential changes.
This indicates that the customer must be present at all times, also in the later stages, as the actual needs of the software develops during the project.

However, verification of the change must be documented for both parties to understand the implication of the change in the development.
A supplier must ask if they have understood it correctly, and thereafter document the customer answer. These demands can be changed at all times, but the customer bears both risk and cost of re-production of the same product.

PS2000 Agile:

Delivery Plan

PS2000 Agile has a strict framework and administration of the delivery plan and its management, which also affect change orders to demand comprehensive work. Kristiansen at Sopra Steria is positive to the comprehensive framework as he finds it to be a great guidance for the delivery execution.
We elaborate the comprehensiveness of PS2000 Agile's delivery plan below.

The guidance in the contract template requires the supplier to periodically report and present the project's status, which contain progress and risk assessment of the software development and implementation, routines, frequency and other terms. In addition, the supplier must as a minimum make a unit test of functionality related to the result of the ongoing Sprint. The parties must thereafter jointly test the part of delivery that is developed, implemented and customized in each Sprint. When ending the development phase, the supplier must conduct a composed integration- and system test to document the delivery functions according to the Solution Proposal, prior to handing it over to the customer.

When the delivery is considered completed, the customer conducts a trial with the supplier’s support, called the Approval Test. The test verifies fulfillment of contractual demands in terms of functionality, performance and other technical tests. This test must be documented in form of process description and approval criteria compiled by the customer in the initial phase of the project. In addition to the Approval Test, the customer has the right to do a trial run of the delivery.

Change Orders

In PS2000 Agile, the supplier is not primarily committed to implement changes to the software development that surpasses the initial estimated solution proposal. If changes are found necessary, the template require comprehensive formal change orders that are considered time-consuming.

The customer can request a change order in form of changes of the scope, delivery content, and change of delivery plan. Similarly, the supplier can issue a change order if the customer demands work or delivery the supplier claims falls outside of the delivery stated in the solution proposal. The supplier must thereby conduct an impact assessment of the delivery, contract price and progress plan, and can demand an added compensation for documented additional cost related to the work of the impact assessment.

Summary of Delivery Plan and Competence and the effect on transaction hazards:

Both templates place the responsibility of the project’s deliveries on the supplier, whereas the Product Owner is responsible providing the supplier with necessary clarifications and verifications. As both templates are based on Iterative Processes, the Product Backlog must at all times be updated with prioritization of features. We recognize this to reduce uncertainty and performance ambiguity as the parties will have a clear vision of what to develop next.

Both contract templates expect changes and therefore allow change orders by both parties. It is found important that the Product Owner has the right competency to verify potential changes of the software development, and thereby change the

Product Backlog accordingly, where technical skills can be crucial. With the right competence, the Product Owner is also able to ensure two Sprints running at the same time, thus avoid stop in the project. Therefore, the right competence copes with uncertainty and performance ambiguity as the Product Owner's skill-set to greater extent ensure successful project delivery and tackle change orders more favorable. A gap in the competence and thereby a stop in the development, creates uncertainty of the project's timeline and cost, and a consequence can also be performance ambiguity in the supplier's next delivery.

SSA-S aligns more with the Agile method by having a flexible delivery plan, thus it forces the involved parties to work under uncertain conditions, where changes are added naturally to the project without comprehensive work needed. Therefore, in addition to the Product Owner, the project group's competency also becomes crucial to lead the project in the right direction based on the customer's need, and to best extent cope with performance ambiguity due to flexible development.

However, one way to cope with uncertainty in the delivery, is by having a comprehensive estimation model of the project, which is found in PS2000 Agile. We find that the contract template requires a much more formal, administrative delivery plan, which affects the change orders' rigidness as well. However, we find the latter making implementing changes troublesome, which is contradictive to the Agile method.

On the other side, changes are expected, and as stated by the informants, how to implement change orders is often dependent on the parties involved. Overall, the extensive documentation of all aspects related to the delivery in PS2000 Agile is recognized as reducing the parties' uncertainty and performance ambiguity.

4.2.7 Termination

	SSA-S and PS2000 Agile	SSA-S	PS2000 Agile
	Similarities	**Differences**	
Termination	Early exit & ongoing exit option for the customer.	Low cost of termination.	High cost of termination.

Table 10: Termination

Similarities among the contract templates

The agile contract templates have an early exit option for both parties, but if the project passes that option, it is only the customer who can terminate the contract. This option is ongoing throughout the whole contractual period.

Kristiansen at Sopra Steria argued that the worst scenario of a conflict is a terminating the agreement, thus in his experience the customer and supplier try to solve disagreements through re-negotiations. He stated that the project normally has lasted for some time, meaning started on the development in the delivery phase, before something incorrect is encountered and disagreement arise. Hence, the parties are already highly involved in the co-operation of the agile project when re-negotiation is needed.

Differences among the contract templates

We find that the main difference between the contract templates is PS2000 Agile's higher cost of termination for the customer than found in SSA-S.

SSA-S:

SSA-S requires both parties' agreement of the solution scope, although it is recognized to be flexible. Therefore, if one of the parties assume a low likelihood of implementing the delivery, the party can opt to exit the agreement prior to the first partial delivery.

After this early exit opportunity, the agreement can primarily only be terminated by the customer, which can be done without hassle at any stage of the project. However, the supplier is able to demand that the customer stops the project, but the customer takes the final decision. Vestre from Difi said that "*an example of this is when the customer is not available and/or present, hence the supplier is not able to do their job. According to the contract, the supplier then has a legitimate reason to demand the customer to stop the project. If the customer does not fulfill his responsibilities as described in the contract, the supplier has the right to exit the contract and get compensation."*

Termination after the early exit opportunity leads to a cost consisting of the customer compensating the supplier for work and cost up until termination, in addition to a fee ranging from 4-6 per cent of estimated total cost.
The cost of exit in SSA-S is perceived low according to Hauge from Statens Vegvesen. The parties also have the possibility to agree upon a self-made termination fee if they do not want to follow the template's proposal.

PS2000 Agile:
In PS2000 Agile, the early exit apply until the contract is signed, meaning that the processes of Need Analysis, Risk Matrix and Solution Proposal are already completed. The parties have therefore already invested in resources and time to complete these stages.

The customer can without further reasons terminate the contract at any stage of the project. If terminating the contract, the supplier is entitled to be compensated for work up until the point of termination, documented direct costs for the supplier tied to the elimination of the contract, and a compensation for the documented loss of profit (limited to a percentage).
Hauge from Statens Vegvesen noted that even though the contract can be exited at all times, they as a customer do not consider this a realistic option due to the related high costs. Also, Hauge noted that the costs linked to termination are already high due to the procurement process of resources and time and are therefore reluctant to re-start the process with a new supplier. This is supported by the notation from Sopra Steria, who always experience a re-negotiation instead of

an early exit of the agreement because investments done in the initial phase of the project by both parties are considered substantial.

Summary of Termination and its effect on transaction hazards:

Both contract templates have an early exit option and an ongoing termination from the customer, but terminating PS2000 Agile carries a higher cost than SSA-S. The latter contract template adheres more to the Agile method by having the early exit option based on the parties' ability to collaborate. If the collaboration does not work optimally, it is easy for the customer to later change supplier if needed. In practice it seems that both customer and supplier find it favorable to solve problems through a re-negotiation instead of termination.

We find that if a disagreement occurs in the initial phase of a project and the parties reach the understanding that the project will not be delivered as wanted, it can be wise to use the early termination option. This will save both the customer and the supplier from engaging and investing more in a project that seems to be complex with low value, which can lead to a bad relationship of opportunistic behavior. We consider the latter to bring uncertainty and risk into the project.

Low cost of terminating a contract can be considered to make it easier to change supplier for the customer, which therefore creates the uncertainty of whether the initial supplier will deliver the finalized project. This creates uncertainty for the supplier in their cash flow and resource allocation. Therefore, the higher lock-in effect in PS2000 Agile is considered to safeguard investments to the project and decrease uncertainty of a termination.

However, as Sopra Steria noted, the normal behavior in the market with agile projects is to re-negotiate the contract agreements, also in the initial phase due to switching costs. If the parties have established a good co-operation, then the likelihood for a successful re-negotiation is larger, which decrease uncertainty for termination.

4.3 Transaction Hazards

In the following sub-sections, we consider the three sub-questions linked to the transaction hazards in our study and give a conclusion to each based on the findings from the seven themes in section 4.2.

The experience and understanding of our interview objects correspond with TCE in how the three transaction hazards are dependent and affect each other. Especially is this concerning the notation that uncertainty does not exist without asset specificity, and that performance ambiguity is a type of uncertainty.

In the interviews, we experienced that uncertainty and risk are considered somewhat the same. In our study we have tried to separate uncertainty as unknown probabilities of events due to lack of knowledge, and risk as specific elements that can be quantified, such as additional cost (Tversky & Fox, 1995; Holton, 2004).

4.3.1 Asset Specificity

SQ1: How does the agile contract templates cope with asset specificity?

When a supplier and customer start the process of collaborating on an agile project, meaning both pre- and post-contract signing, asset specificity occur. There is an immediate investment in the project through human resources by allocation of employees, time and competency, and the parties' knowledge of each other, and in some instances through tangible assets.

Different from other ICT projects is the agile contracts' need for more extensive customer involvement with high competency. The Product Owner must execute good leadership in the agile development project and have technical skills regarding the software being developed, which we consider being asset specificity. The supplier also joins the project group with its resources of expertise on the software to be developed.

Therefore, both parties have the interest to safeguard one's investments, and some firms wants to have a framework that locks the co-operation and enables the Agile method.

The interview objects at PROMIS, Sykehuspartner and Sopra Steria noted that if the parties are locked to a contract, they feel more secure about own investment, thus more willing to invest further in the relationship and the project.
Especially does safeguarding concern the supplier's uncertainty of whether the customer will terminate the agreement and change supplier.
Sykehuspartner reasoned that a supplier would rather use its best resources in a contract they are more secure will harvest from their investments, as opposed to a contract they find unsecure.

We find that PS2000 Agile's more binding contract with higher cost of termination to a larger extent copes with safeguarding asset specificity than what SSA-S does.

Additionally, asset specificity is also protected by different factors coping with uncertainty, which we elaborate in the sub-section below.

4.3.2 Uncertainty

SQ2: How does the agile contract templates cope with uncertainty?

As the agile contracts requires the parties to make investments, the project and the relationship will from start to finish being considered uncertain as the parties potentially can lose their investments. This is consistent with the notations of the informants, which recognized uncertainty as a natural part of ICT projects.

The Agile method calls for flexibility in the software development project as it is welcoming changes with the aim for best possible value outcome, thus the end-result should be an evolving process. Uncertainty therefore exist, but both contracts also have aspects lowering uncertainty, dependent on the degree of agility implemented.

When considering how the contracts are coping with uncertainty, one can evaluate rigidness versus flexibility, where the latter is more aligned with the Agile method. Our observations note that rigid formal contracts lower uncertainty more than flexible contracts that favors co-operatorial problem-solving, which is aligned with TCE (Gilson, Sabel, & Scott, 2009; Jiang, Gao, Bao, & Jiang, 2012). The latter is thus moving towards relational governance, which is important when implementing an agile method. This means that one can view how the standard templates are coping with uncertainty from different angles.

One way to cope with uncertainty is thus to have a rigid contract with an agreed end-result that the supplier must deliver on, and where elements of risk are pre-discussed and delegated (PS2000 Agile). The other option is to have close co-operation where both parties expect changes during the project, are positive to the uncertainties surrounding them, and co-work until the customer is happy with the unknown end-result (SSA-S).

It seems that the above differences in coping with uncertainty highly depends on the involved parties' competency and how they choose to co-operate during development, delivery and problem-solving. Within development and delivery are project status updates and transparency of knowledge-sharing, which we find copes with uncertainty as it ensures both parties' alignment to the project's status. Additionally, the parties' co-operation can create a close relationship among the human resources representing the parties, thus trust can more easily be built which in turn also reduces uncertainty.

Based on the above findings, we consider competency and co-operation to be crucial to cope with uncertainty.

4.3.3 Performance Ambiguity

SQ3: How does the agile contract templates cope with performance ambiguity?

As agile delivery requires a large degree of resource investment to enable the software development, it is important to cope with performance ambiguity to avoid further resource spending and hold the planned time line of the project.

Firms and people with different backgrounds and interests in a project can lead to occurrence of misunderstandings and spotting observations. This affects the development and delivery of the software by creating performance ambiguity. To cope with this, it is important to be aligned on the Solution Proposal, and to have good project management and co-operation throughout the project. The aspects that copes with performance ambiguity are linked to those found to cope with uncertainty, which demonstrates that uncertainty and performance ambiguity are related in agile projects.

The largest difference between the two agile contract templates in regards of coping with performance ambiguity is the more rigid solution description in PS2000 Agile. Overall, we find that the Agile method and its iterative process with Product Backlog and Sprints implemented in both agile contract templates cope well with performance ambiguity by decreasing it.

Conclusion of the three Transaction Hazards

Both customer and supplier need to safeguard their respective investments as agile projects has uncertainty and performance ambiguity can occur, and the aim is to avoid opportunistic behavior in the relationship. This stresses the importance to find a contract framework with optimal balance of rigid formal context and relational governance that matches the transaction attributes concerning the project and enables an agile software development.

Through the analysis of the transaction hazards, we find that PS2000 Agile’s comprehensive and rigid contract framework copes well with the transaction hazards, but that the same elements are also what makes the contract less agile. Thus, it seems to be a tradeoff between the Agile method and coping with the transaction hazards.

Overall, concerning both contract templates, we find them to require asset specificity of human resources, but that the investments itself copes with uncertainty through high competence in the project group and Product Owner.

With the project group, the agile contract templates require co-operation between the parties during the development and the delivery phase. This also copes with uncertainty, and lower performance ambiguity, as the project group will more easily be able to develop a successful value outcome, which is the most important objective in the Agile method.

5. Discussion

In this chapter of our master thesis we discuss the findings from the Literature Review and Analysis of the Agile Contract Templates. The discussion deliberates how the agile contract templates have implemented the Agile method, and how this further affects how the templates cope with the transaction hazards.

This leads us to answer the research question of our study:

> *"How does agile contract templates cope with transaction hazards in ICT projects, and how does that affect when to use the contracts?"*

5.1 The Agile method versus Agile Contract Templates

We find that the agile contract templates differ in the degree they are aligned with the Agile method, and that neither of them has fully implemented all the values and principles of the Agile Manifesto. This is also noted by the informants, who stated that when using the Agile method, a large framework, such as SSA-S and PS2000 Agile, should not be required. Instead, the creators of the contract templates have tried to create frameworks that enables use of the method as a base, however to different extents, which we elaborate below.

PS2000 Agile is recognized as a comprehensive contract known for its demanding administration, where several of the informants noted the need for additional consultants (e.g. PROMIS consultants or lawyers) to understand the framework and ensure compliance with its jurisdictions. With different rigid guidelines and commitments in PS2000 Agile, it is recognized to prevent parts of the Agile method.

PS2000 Agile is created with emphasis on ensuring a binding contract, where potential risk scenarios are assessed through the Risk Matrix and distributed to a responsible part, where it can be incorporated to the price estimates. Thus, PS2000 Agile requires early assessment of risk and decision on the price model in the initial phase of the project, which implies knowledge of the final solution. Hence, PS2000 Agile is found contradictive to the Agile method where the final solution is unknown, and risk must be managed as an ongoing process.

The project's development phase in both SSA-S and PS2000 Agile is to be executed through iterative processes based on the Agile method, which provide flexibility in the software development. However, in PS2000 Agile, there is a commitment towards the supplier to deliver on the agreed Solution Proposal and pre-specified end-result. This limits the project's development to implement changes that are beyond the scope, as the contract template require comprehensive change orders towards the supplier. We find this contradictive to the Agile method where the end-result is to be developed during the project.

SSA-S is to greater extent enabling the use of the Agile method, which is also consistent with the opinion of the creators. This is due to its foundation of flexibility, open-ended format and welcoming change orders as the development moves along, to ensure best possible value outcome.

Hence, we consider SSA-S to be more aligned with the core values and principles in the Agile method than PS2000 Agile.

5.2 The Agile Contract Templates versus Coping with Transaction Hazards

Through the analysis where we consider how the agile contract templates cope with the three transaction hazards, we find that the Agile method in many ways require asset specificity and uncertain conditions but are coping well with performance ambiguity. This is consistent with studies reporting that factors leading to successful ICT projects are aligned with the core values in the Agile method of how to drive projects, but that there will be uncertainty concerning the project (Dönmez & Grote, 2017; Jørgensen, 2015).

However, we find that how the agile contract templates cope with the transaction hazards also depends on the customer and supplier involved in the project, and how they comply with the contract templates' requirements of human resources. Both PS2000 Agile and SSA-S emphasize co-operation in its templates and have requirements of human resources from the customer and supplier to be in the project group. Additionally, we identify that there is a special demand for competency at the customer-side to enable a successful agile development project. However, the contract templates differ in their rigidness of how the agile project should be executed.

Based on our analysis, we therefore consider project management, relational governance and customer's competency to have the highest effect on the transaction hazards as having these established will determine how the project is executed, and thereby affect the success of the agile project.
We find that independent of which contract framework the project is complying to, having a higher degree of the above-mentioned factors implemented in the agile project will to a larger extent decrease uncertainty and thereby performance ambiguity, and influence safeguarding asset specificity.

This is summarized in Figure 3 below:

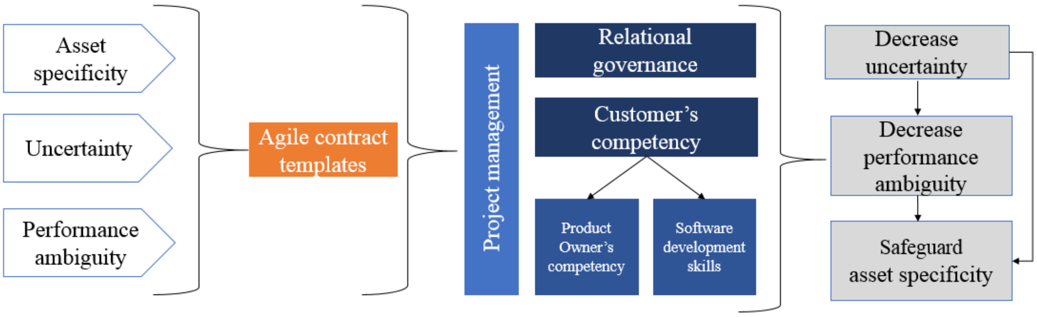


Figure 3: Research findings

In the following sections we elaborate each element.

Project management

Proper execution of project management is important in all projects, as it can affect the direction and success of the outcome. Thus, we find this to be a foundation also in an agile development project, and without it, it is not possible to drive the iterative processes in the delivery plan. A well co-operated project group, led by the Product Owner who manage the development to meet the customer's need, is therefore found to cope with the transaction hazards.

Relational Governance

The focus of building relationship with the stakeholders is realized in today's business culture. Still, the choice of implementing the Agile method to a software development project demands an even closer relationship between the parties. Because SSA-S and PS2000 Agile have implemented the Agile method to some extent to its contract templates, the contracts are considered trust-based and highly dependent on co-operation.

Therefore, we find it important that the parties in the contract builds a relational governance as a foundation in the agile project. As noted by Sopra Steria, there are high investments from both parties and the agile project is challenging, thus the co-operation must be capitalized where competency and relation are built jointly. This means that both the customer and supplier must place value on the relational exchange, and invest and delegate human resources to the co-operation, to create mutual trust and solidarity between the parties in the agile project.

Therefore, we find that established trust and solidarity will decrease the uncertainty of opportunism, and the parties are more motivated to be adaptable when needed and to share information, which aligns with the findings of Heide and John (1990). This is positive at times when obstacles in the software development occurs, where trust serves as a good foundation for enabling a solution for both parties to cope with the problem.

We find that since SSA-S leans more towards the Agile method, there is an expectation from the parties to work under uncertain conditions. Thus, they are more likely to cope with occurring problems together as a part of the project, not as something adding to it. As the responsible part of potential risk scenarios is not pre-decided in SSA-S, relational governance can be critical.
However, based on the interview objects' experiences, they all expects changes and problems occurring while driving an agile software development project, and that how these are solved often depends more on the parties involved and their established co-operation, rather than the guidance from the contract template used.

Hence, if solidarity is established through relational governance, the parties will consider the project as a shared responsibility, and hopefully solve problems with less opportunism and rather with the aim of a successful project.

Customer's competency

The agile contracts differ from other standard contracts as the method in them requires a high degree of participation from the customer throughout the whole project to act as the Product Owner. If the needed competence required for the agile project is not sufficient, it will influence's the Product Owner's operations towards the supplier, and thereby create uncertainty and performance ambiguity for the involved parties.

We find that if the customer has the needed competency for the agile project and the software to be developed, this copes with uncertainty as the customer is better able to understand the supplier's Solution Proposal, and thereby evaluate the complexity and risk of the software development. Thus, during the supplier's software development and delivery, the customer can also provide quick clarifications and verifications when needed. This means that agile projects can be executed more efficiently and in a shorter timeline, and in the direction to meet the customer's need of the software developed.

We consider the customer's Product Owner's technological competence to have great impact on the transaction hazards. Also, we find that the customer can match the supplier's expertise of software development skills, and thereby co-work during the agile software development, which we consider copes with the transaction hazards. We elaborate this below.

Product Owner's Technological Competence

We recognize that the agile contract templates require the Product Owner to possess profound competency of the non-functional demands of the customer's need in order for the software to be successfully developed.

Based on findings from the analysis, it seems that the greater the competency of the Product Owner, the more it decreases uncertainties and performance ambiguity. This is related to the Product Owner's technical understanding, which will enable him or her to evaluate the complexity of the project. Thus, it enables frequent and rapid clarifications and verifications to the supplier, which in turn enables the development to move forward in the correct direction.

Recognized by the interview objects, the correct high-level competence is sometimes difficult to retrieve from the Product Owner, which can imply that customers are withholding investments in the specialized competence and resource allocation to avoid asset specificity. This can be troublesome as both agreements gives the supplier the right to demand that the customer puts the project on hold if the Product Owner does not comply to its commitments of clarifications and decisions. However, both contract templates force the customer to continue paying the supplier while waiting for verifications to continue the software development, which is considered to cope with the lack of involvement from the customer-side. Hence, if the customer does not make the required investment in resource allocation of a competent Product Owner, the customer is punished through "unnecessary" payments to the supplier even though the development is on hold.

Noted from Vestre at Difi, this is a mechanism to involve the customer to a higher degree than what is normal in ICT projects. The high-involvement and competency foster more work for the customer, but in turn creates a closer relationship among the parties where the customer can manage the progress of the software development in the correct direction. Therefore, this helps to cope with uncertainty and create a successful project.

Thus, we find that due to how agile projects are driven, the customer will highly benefit from investing in a competent Product Owner who has the needed technological skills in order to communicate appropriately with the project group and the supplier's developers. Hence, coping with uncertainty and performance ambiguity in the agile software development.

Software Development Skills

We find it important that the customer has software developer representatives, either internal or a hired 3rd part, who are able to evaluate the Solution Proposal provided by the supplier. This is found beneficial due to several reasons.
First, with internal software skills the customer will be able to understand the complexity of the project, thus assess the supplier's judging of the complexity, needed work hours and price estimation. Second, it is beneficial for the customer's value outcome if they are able to manage the development project in the desired direction, thus enable co-operation that to a greater extent ensure that the end-delivery will be satisfactory for the customer. The latter can easier be done by having internal software developers who can co-work with the supplier's developers.

This is consistent with the trend found in Industry 4.0 of internal technological competency, which posit a higher call for in-house teams of developers.
This type of investment is found at several Norwegian entities, e.g. NAV (the Norwegian Labor and Welfare Administration). NAV is expanding their IT department with internal software developers, aiming to match the expertise of the suppliers in order to closely co-operate and efficiently drive agile projects (Jørgenrud, 2016; Østvang, 2016). Further, expansion of a company's workforce is a good investment in asset specificity concerning digitalization.
Hence, this trend aligns with our finding of co-operation and customer's resources having software development skills matching the supplier's to be important.

The customer's investment in internal competency of software developers is a way of internalizing exchange where the firm is choosing a more internalized governance structure, thus gaining more control of the needed competency.
By having the internal competence, the firm safeguard its asset specificity in the software development project against uncertain opportunistic behavior that can be a threat from external developers (Ghosh & John, 1999).
Based on the above, we find that having high competency in software development at the customer-side copes with uncertainty in the project as it enables closer co-operation between the parties' resources. The customer and supplier can therefore co-work to create the codes of the software, which we also consider decreasing performance ambiguity where the parties are more aligned on

the technological level of features needed in the software. Thus, an internal software developer can also help the Product Owner during the deliveries.

5.3 Managerial Implications and Decisions

In the following section, we present the managerial implications and decisions of this study. The sub-section of managerial implications demonstrates the results of our discussion related to actions, whereas the sub-section of managerial decisions will provide insights to operating decisions.

5.3.1 Managerial Implications

Aligned with Jørgensen's (2015) report, our study argue that the choice of project method can impact the transaction hazards and the delivery outcome, thus the choice of agile contract template has managerial implications.

The data analysis in this study has shown that the agile contract templates differ in the extent they enable the parties to drive a true Agile method relative to the Agile Manifesto's values and principles. Thus, the customer, or both parties, must either first choose which contract template is desired and thereafter consider how the project will be affected accordingly, or it must first choose how agile the project is supposed to be, and thereafter choose the contract template.

Research has found that when choosing a governance form, one must strive to find an optimal balance between rigid formal contract and relational governance that aligns with the transaction attributes concerning the project where a fit can create a complementary effect (Sande & Haugland, 2015; Poppo & Zenger, 2002). If the project aims to enable software development by the Agile method, the contract must accordingly be flexible and adapt to changes the parties find necessary for an optimal value outcome. Too rigid contracts can create obstacles for the parties to adapt when needed, which can affect the outcome of the agile project. On the other side, less rigid contracts can create uncertainty concerning the other party's actions when changes are needed in the agile software development.

We find that independent of the agile contract template chosen, execution of an agile development project is dependent on project management, relational governance and the customer’s competency.

Both the customer and supplier must as a bare minimum have knowledge and understanding of how the Agile method works. Further, the Agile method requires close co-operation and the agile contracts are trust-based, which implies that relationship building is highly necessary. This is especially important when the contract leans towards flexibility and more agile development, such as in SSA-S. Thus, both the customer and the supplier must evaluate whether they have available human resources to build the necessary relationship to ensure mutual understanding and trust between the parties, and the competency to build a strong co-operation and execution in the project group.

5.3.2 Managerial Decisions

The use of the agile contract templates requires a great resource allocation from the customer of human assets, their competence and time to comply to the required customer involvement. This demand is also recognized as a key difference between agile contract templates and traditional procurement of software, where the customer orders a solution from the supplier who undertake the whole delivery and risk.

Further, when choosing to drive an agile project and which agile contract template to use, the customer must evaluate its ability and willingness to be responsible for risk scenarios and uncertainty concerning the development project. The choice should be based on the customer’s ability to affect the risk elements, and their internal competence of agile project execution and software development.

If the customer has little impact on the risk scenarios, they should choose PS2000 Agile due to its Risk Matrix, and as an insurance pay the supplier to handle potential obstacles. If the customer is able to affect the risk scenarios and thereby reduce uncertainty, and more willing to bear the risk themselves, the customer should choose SSA-S.

To summarize, we find that both the customer and supplier must take a stand to their ability to comply to the requirements in the agile contract templates, where especially the demand for human resources is considered high. Further, the choice of agile contract template should be aligned with the transaction attributes, aiming for an optimal balance of following a rigid formal contract to cope with uncertainty, and a relational governance to enhance flexible co-operation.

5.4 Limitations & Future research

In this section we consider the identified factors that can have influenced the results of this study. Thereafter we provide examples of further research regarding the topic of agile contracts and transaction hazards.

5.4.1 Limitations

Possible limitations to this study are recognized as the sample, the language used when conducting the interviews, and generalizability. Each theme is presented in the following sub-sections.

Sample

Sample selection

The informants used for this study are drawn from a non-random method called strategic selection which entails that we as researchers have consciously chosen the informants, as previously described in sub-section 3.4.3.2. This could have led to biased selection, which can lead to a reduction of generalizability of our findings. However, qualitative methods are concerned with in-depth understanding of the phenomena rather than generalizability (Patton, 1987; Dworkin, 2012).

The sample withdrawn is chosen due to their roles, company and experience, which entails insight of how the agile contract templates are utilized.
Thus, the selection method is required in order to get informants that provide insight that enables us to answer to the research question.

Sample size

Qualitative studies require less participants than quantitative, but the question of “how many” is not easily decided, but a great portion of the debate argue that 5-50 is adequate (Dworkin, 2012). The sample in this study can be considered somewhat small with five interviews. However, in three of these interviews, there were two informants from the same company present. In practice, this means a total of eight informants.

Many argue that the point of saturation is the most important to consider in regard to sample size, which concerns the point where the data collection does not yield new or relevant information related to the research (Dworkin, 2012). We find that there is a high degree of consensus between the informants, where the experiences from one informant more often than not coincided with the experiences of the other, regardless of their role. Examples of this is present in the chapter of analysis. We therefore believe that the addition of more interview would not affect our findings, thus arguing that the sample size is representative.

Additionally, we once again pinpoint that our data set is two-folded, where the contract template represents data source 1 and the interviews represent data source 2. We believe that this contributes to the adequacy of a relative low number of informants.

Interviews Conducted in Norwegian

All interviews were held in Norwegian and presented in English in this thesis. This raises the issue of meanings being "lost in translation", where we acknowledge that language is both complex and contextual, and the true meaning of a statement might not translate well word-by-word.

However, both of us as researchers are native Norwegians who speak the language fluently. This means that we understand the meaning and the context the examples were explained in to full extent and could adjust the wording for translation to fit the statement correctly. Additionally, all informants were given full transcriptions from the interviews in order for them to correct possible misinterpretations and/or mistranslations.

Generalizability

Generalizability concerns the extension to which the findings of research can be conceptualized and be significant for the population (Steendam & Bergh, 2012).

As the goal of qualitative research is often related to small samples providing in-depth information, generalizability is not easily obtained.

Nevertheless, we want to comment on the generalizability in our study.
Our work leads us to believe that many aspects make transactions differ, e.g. complexity, scale and firms involved. Thus, we can therefore not claim that the findings are representative for all transactions of ICT projects using an agile method. Therefore, our conclusion will not be an absolute general statement of how the agile contracts cope with the transaction hazards. However, discussion and conclusion are based on the data analysis conducted in this interpretive research, and as Williams (2000) note, one can claim that this will produce a certain extent of *moderatum generalization*. This means that through constant impressions we make judgements about the truth that is included in our evaluation.

As we argue that our sample is representative, we assume that the findings are generalizable for the three transaction hazards in regard to agile contracts for ICT projects in Norway. However, we cannot assume that our findings can be transmitted to other development projects, as this can have other important implications not included in this study.

5.4.2 Future Research

As a result of the limitations, this master thesis might serve as a good starting point for future research of the topic.
As noted, it is unsure how generalizable and transferable the study is. Thus, a good starting point for future research could be to include other transaction hazards of TCE found important and see how these are coped with by the contract templates. In addition, it can be relevant to consider the phase before the contract signing, which is the procurement process. New procurement methods are being developed and implemented in the public sector to enable innovative projects, and this has impact on the agile method and the transaction in ICT projects.

Dynamic Purchasing System

There is a new procurement method in the market, called Dynamic Purchasing System (DPS), which is of interest to public sector companies, and already used by some, for example Ruter AS (Company for public transport in Oslo and Akershus, noted by Sopra Steria during the interview).

DPS is a procurement tool similar to an electronic framework but differ in the extent that suppliers are able to join at any time, and the procurement must run as a completely electronic process (Procurement For All, 2018). Most of our informants mentioned this procurement form which is relatively new, and we expect that this will influence how many of the public companies will conduct procurement moving forward. This means that the agile contract templates will be affected by the procurement phase and may get competition. Future research could assess how our findings of transaction hazards are transferable to DPS, and how the procurement model affects the use of the agile contract templates.

SSA-B

Related to the demand for customer’s competency in agile projects, is the statement from some informants that if the internal competency is high with software developers and how to drive an agile method, they would opt to use SSA-B (Difi’s contract for Assistant Agreement). Thus, simply buy consultant heads. As there is a trend of developing internal in-house competency, we believe that SSA-B could also pose a threat to the use of agile contract templates. It could therefore be interesting to further study the implications of driving an agile project with SSA-B in relation to SSA-S and PS2000 Agile, and the transaction hazards.

Quantitative Research

It could also be interesting to study the topic using quantitative method.

This can be used to find which of the agile contract templates works better in the market by “hard” measures as degree of success, time-to-market, number of optimal sprints, degree of time and budgetary transgression.

6. Conclusion

The purpose of this study has been to examine how agile contract templates copes with transactions hazards in software development projects, and further how that affects when to use the agile contracts.

We consider PS2000 Agile as a comprehensive framework that safeguard asset specificity the most through its lock-in effect of supplier-commitment and assessment of risk, which decreases uncertainty. By the commitment to an agreed end-result, it is considered easier to avoid performance ambiguity. SSA-S is found more flexible and open-ended towards the final solution, and with less commitment to the agreement, thus does not safeguard asset specificity, or reduce uncertainty and performance ambiguity in the same extent. Thus, we consider that there exists a tradeoff between the Agile method and coping with the transaction hazards. However, our findings demonstrate that neither of the two agile contract templates have implemented the Agile method to full extent but are rather based on the method. Hence, the contracts enable both agile project development and coping with the transaction hazards.

We find that the customer and supplier should aim to build relational governance, where the customer should have a Product Owner with technical competence and internal software developers. This will decrease uncertainty of opportunism as the parties have a trust-based relationship, reduce performance ambiguity through close co-operation, thus safeguard asset specificity.

Thus, we conclude that independent of the agile contract template being used for a software development project, well-executed project management with relational governance and high customer competency, will cope with the three transaction hazards to the highest degree.

We further conclude that the choice of when to use the agile contract templates depend upon the customer’s evaluation of internal competency. A firm that tolerates a high degree of uncertainty and aspire to work as agile as possible, should use SSA-S, whereas a firm that is risk-averse and favors a rigid framework should opt for PS2000 Agile.

7. List of Figures and Table

7.1 List of Figures

Figure 1: Manifesto for Agile Software Development 10
Figure 2: Illustration of differences between Waterfall development and 13
Figure 3: Research findings 77

7.2 List of Tables

Table 1: Informants 31
Table 2: Explanation of terms 43
Table 3: Contract overview 44
Table 4: Need Analysis.. 45
Table 5: Solution Proposal and Scope 46
Table 6: Risk Analysis.. 49
Table 7: Price model... 53
Table 8: Project group 56
Table 9: Delivery Plan & Competency.. 59
Table 10: Termination 67

8. Appendices

Appendix 1: Agile Manifesto

Principles behind the Agile Manifesto
Reference: (Beck, et al., 2001)

1) Our highest priority is to satisfy the customer
through early and continuous delivery
of valuable software.

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

3) Deliver working software frequently, from a
couple of weeks to a couple of months, with a
preference to the shorter timescale.

4) Business people and developers must work
together daily throughout the project.

5) Build projects around motivated individuals.
Give them the environment and support they need,
and trust them to get the job done.

6) The most efficient and effective method of
conveying information to and within a development
team is face-to-face conversation.

7) Working software is the primary measure of progress.

8) Agile processes promote sustainable development.
The sponsors, developers, and users should be able
to maintain a constant pace indefinitely.

9) Continuous attention to technical excellence
and good design enhances agility.

10) Simplicity--the art of maximizing the amount
of work not done--is essential.

11) The best architectures, requirements, and designs
emerge from self-organizing teams.

12) At regular intervals, the team reflects on how
to become more effective, then tunes and adjusts
its behavior accordingly.

Appendix 2: Interview Guides

Below is Interview Guide 1 and 2.

Interview Guide 1

Interview Guide 1 was used in interviews with Difi and PROMIS.

The contract standard name (PS2000 Agile/ SSA-S) was changed according to which object was interviewed.

Part 1: Introductions of the researchers and presentation of the research question. (5 minutes)

Part 2: Fact/Introductory questions (10-15 minutes)

1. What characterizes a good contract for IT system implementation?
2. What was the background for the development of PS2000 Agile/SSA-S?
    a. Was it requested by the market?
3. What was the background for developing the contract in cooperation with both customers and suppliers?
    a. What were the pros/cons?

Part 3: Main part (25-35 minutes)

4. How is PS2000 Agile/SSA-S formed in regard to asset specificity?
    a. How does the contract secure the specific investments?
    b. Pros/cons/examples
5. How is PS2000 Agile/SSA-S formed in regard to uncertainty?
    a. How does the contract secure for uncertainty?
    b. Pros/cons/examples
6. How is PS2000 Agile/SSA-S formed in regard to performance ambiguity?
    a. How does the contract secure for performance ambiguity?
    b. Pros/cons/examples
7. Are there any missing aspects/”open holes” in the contract relative to the 3 transaction hazards?
8. How is PS2000 Agile/SSA-S been accepted by the market?
    a. Examples?
9. Which contract magnitude is required for PS2000 Agile/SSA-S to be well-suited?

10. PS2000 Agile/SSA-S requires a close co-operation between the customer and the supplier, how does this close relation substantiate collaboration and flexibility?
11. How is PS2000 Agile/SSA-S used in practice?
    a. Do the parties follow the contract step-by-step or does it function more as input for the parties?
12. Is PS2000 Agile/SSA-S as beneficial for the supplier as for the customer?
13. What do you believe is the most important aspects of PS2000 Agile/SSA-S?
14. Could you provide some examples of successful/unsuccessful use of PS2000 Agile/SSA-S?
15. What are the benefits of using PS2000 Agile/SSA-S versus SSA-S/PS2000 Agile?

Part 4: Ending questions (10 minutes)

16. Is there something you consider important for future IT process contracts which is not included in today's PS2000 Agile/SSA-S contract standard?
17. Is there anything you would have asked if you were us?

Interview guide 2

Interview guide 2 was used in interviews with Statens Vegvesen, Sykehuspartner and Sopra Steria.

Part 1: Introductions of the researchers and presentation of the research question (5 minutes)

Part 2: Introductory questions (10-15 minutes)

1. What characterizes a good contract for IT system implementation?
2. How do you perceive SSA-S and PS2000 Agile?
    a. Main aspects
    b. Main differences
    c. Pros/cons
    d. When does one fit more than the other one?

3. What are the benefits of using SSA-S versus PS2000 Agile, and vice versa?
4. How are SSA-S and PS2000 Agile used in practice?
    a. Do the parties follow the contract step-by-step or does it function more as input for the parties?
5. How are the two different contracts in terms of being more beneficial for the supplier or the customer?
6. Could you provide some examples of successful/unsuccessful use of SSA-S and PS2000 Agile?

Part 3: Main part (25-35 minutes)

7. How do you regard asset specificity in IT system implementation?
    a. What asset specificities do you see/find occur in IT projects?
    b. Are there any misalignments here between customer and supplier?
    c. How do you find SSA-S and PS2000 Agile coping with asset specificity (protecting them, create mutual understanding)?
    d. How do you perceive the ability to safeguard the specific investments in terms of asset specificities in the contract templates?

8. How do you regard uncertainty in IT system implementation?
    a. What uncertainties do you see/find occur in IT projects?
    b. Are there any misalignments here between customer and supplier?
    c. How do you find SSA-S and PS2000 Agile coping with uncertainty (decrease, increase)?
    d. How do you perceive the ability to be flexible in terms of uncertainty in the contract templates?

9. How do you regard performance ambiguity in IT system implementation and the agile delivery?
    a. When do you see that performance ambiguity occur?
    b. How do one solve such performance ambiguity?
    c. How do you find SSA-S and PS2000 Agile coping with performance ambiguity (decrease, increase)?
    d. How do you perceive the ability to measure performance in terms of performance ambiguity in the contract templates?

10. Are there any missing aspects/ ”open holes” in the contract relative to the 3 transaction hazards?

Part 4: Ending questions (10 minutes)

11. Is there something you consider important for future IT project contracts which is not included in today’s SSA-S and PS2000 Agile templates?
12. Is there anything you would have asked if you were us?

Appendix 3: Informant Request

Below is the first mail sent to our informants to initiate contact and request participation for interview.

The e-mail was sent in Norwegian but translated to English for the purpose of this thesis.

“Hi *[name of informant],*

Marte Hagelund, and myself, Helen Dyb, are two master students at BI Oslo.

This spring we are writing our master thesis, where we want to look at the differences between SSA-S and PS2000 Agile.
We base our theory on Transaction Cost Economics and aim to analyze how the contract templates copes with three elements of this theory: asset specificity, uncertainty and performance ambiguity.

In regard to this and our qualitative study, we find it very interesting to have an interview with you, to get your viewpoint and input on this matter and ask some specific questions.
Do you have the possibility to spend 1-hour meeting Marte and me in the nearest future?

Both work 50% along with our studies, so the best day for us is Thursdays, but we are flexible all other days as well.

We look forward to hearing from you.

Best regards,
Marte Hagelund and Helen Dyb”

Appendix 4: Quotation-check

The following e-mail is a quotation-check and was sent to the interview objects for approval. The participants all received the same text in the e-mail, with an individual attachment of their statements and the quotes that we would include in this thesis.

Except from Sopra Steria, all were interviewed during spring. The deadline in the e-mail text is therefore correct for all others, and Sopra Steria had a later deadline.

The e-mail was sent in Norwegian but translated to English for the purpose if this thesis.

"Hi *[name of informant],*

We want to once again thank you for taking the time to participate in the interview. Your contributions were very valuable and helps us writing this master thesis.

As mentioned, we are now sending you a document with your quotes that we find relevant to further use in our study. In the document you will find that we have written a summary from the transcribed audio recording. We have presented the different themes, thereafter your contributions, in order for you to see that nothing is taken out of context and thereby secure validity and reliability in this master thesis.

Hence, this is the document we further will use to extract information from.

We have set the deadline for feedback/approval to 01.07 at 12.00 pm.

Best regards,

Helen Dyb & Marte Hagelund"

Appendix 5: Analysis Process

Excerpt from the analysis process

In this appendix, we will demonstrate how our analysis process was done.
By following Askheim and Grenness' (2008) approach to qualitative data analysis explained in section 3.7, color coding can be used to categorize, code and structure the findings from our data sources.

This is done for both data sources. We will below show an excerpt from one of the interviews and one of the contract templates. However, the process is similar for all the interviews conducted and the other contract template.

Color codes:
Pink: How does the agile contract templates cope with asset specificity? – Sub question 1
Yellow: How does the agile contract templates cope with uncertainty? – Sub question 2
Green: How does the agile contract templates cope with performance ambiguity? – Sub question 3

Example 1: Interview with Jørgen Petersen at PROMIS

Main interviewer: *In our experience, both SSA-S and PS2000 Agile has high demands for the customers competency.*
Petersen*: "Yes, this is another element. The customer must be more participating and try to match the supplier in a higher degree and need to communicate with the supplier in a higher degree".*

Main interviewer: *"...leads to a high degree of uncertainty with an agile contract. Would you say that PS2000 copes better with this due to the Risk Matrix?"*
Petersen: *"with the Risk Matrix you assess the risk and create awareness. If the customer states that there is risk, and the supplier is not able to eliminate this, the supplier must price it. And the customer can adjust by doing something with the risk"*

Quote from a longer example of a case with a customer

Petersen*: "The solution description is PS2000 Agile aims to give a coherent picture, both the supplier and the customer, of how the solution will look at the end"*

Example 2: SSA-S Contract template

The below examples are directly cut out from the contract template and pasted in full form. In order to demonstrate the context, the whole section is included.

6. THE DUTIES OF THE CUSTOMER

6. 1 RESPONSIBILITIES OF AND CONTRIBUTIONS BY THE CUSTOMER

The Customer is responsible for having described the purpose of the procurement and its requirements and needs, in Appendix 1, in a clear manner, as a basis for the performance of the Contractor, and for actively participating in the implementation of the deliverables in conformity with the software development method in Appendix 6.

The Software and other equipment with which it shall be compatible, and any Software and equipment that shall be used during the development and testing of the deliverables, are described in Appendix 3. If it is stated in Appendix 2 that the technical platform of the Customer needs to be upgraded, the Customer shall itself ensure such upgrading, unless otherwise stipulated in Appendices 1 and/or 2.

The Customer shall actively contribute to facilitating the performance of the Contractor's obligations under this Agreement, including ensuring that clarifications and decisions are made, so that the software development can be performed in accordance with the ReleasePlan, cf. clause 2.2.2.

2. PERFORMANCE OF THE DELIVERABLES

2.7 EXIT, CANCELLATION AND TEMPORARY SUSPENSION

2.7.1Exit prior to acceptance testing of first Release

If one of the parties believes that it is improbable that the deliverables will be executed in conformity with the Agreement, it may decide to withdraw from the Agreement until the acceptance test for the first Release begins (Exit). The Exit arrangement is in addition to the ordinary cancellation provision in clause 2.7.2.

The party that has taken the initiative to the Exit must notify the other party in writing, presenting the main lines of the challenges. The other party must respond to the notice within five (5) Working days. Together with the notice, the party that has taken the initiative to the Exit must convene a meeting between the parties where they discuss the options for continuing the project. Each party may demand participation of an independent expert appointed according to the rules in chapter 16. The party that has taken the initiative to the Exit must pay the consideration for the independent expert.

If the parties do not agree to continue the project within thirty (30) days of the written notice in accordance with the paragraph above, the party that demands the Exit may withdraw from the Agreement with immediate effect.

Unless otherwise agreed in Appendix 7, in connection with the Exit initiated by the Customer, the Contractor will receive consideration for the work that it has already performed, and any other necessary direct expenses associated with licences, equipment and other goods procured for the project before the date of the Customer's Exit. In that case, all rights to such licences, equipment and other goods must be transferred to the Customer. If the Exit was initiated by the Contractor, the Contractor may claim reimbursement of direct expenses as mentioned if the Customer wants to take over such licences, equipment and goods. If the Exit was initiated by the Contractor, the Contractor cannot claim consideration for work that has already been performed.

2. PERFORMANCE OF THE DELIVERABLES

2.1 SPECIFICATION, TESTING AND FIELD TESTING OF THE RELEASE

2.3.1 Detailing and specifying the Needs Specifications

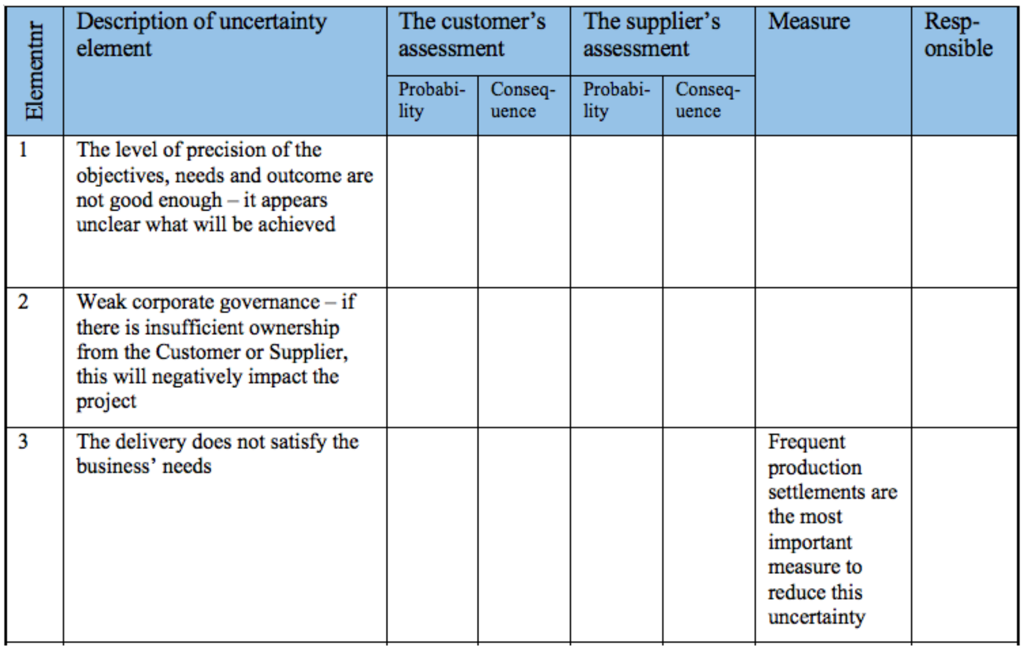
The parties must collaborate on detailing and specifying the Needs Specifications and the requirements in Appendix 1 in accordance with the software development method described in Appendix 6. As part of the work, the Customer should define the Pass/fail criteria that must be met in order for the functionality developed to be accepted.

Appendix 6: The Risk Matrix

Below is an excerpt from the extensive Risk Matrix included in PS2000 Agile, mentioned in section 4.2.3. The full matrix consists of 14 elements, where we demonstrate the first 3 to show the form and function. As it is presented, it is to be considered a check-list that needs concretization and adjustments to each specific delivery.

The excerpt is made and translated from Norwegian by the authors of this thesis.

PS2000 Agile must be purchased, thus due to copyright reasons we are not able to provide a reference for the Risk Matrix.



Elementnr	Description of uncertainty element	The customer's assessment		The supplier's assessment		Measure	Resp-onsible
		Probabi-lity	Conseq-uence	Probabi-lity	Conseq-uence		
1	The level of precision of the objectives, needs and outcome are not good enough – it appears unclear what will be achieved						
2	Weak corporate governance – if there is insufficient ownership from the Customer or Supplier, this will negatively impact the project						
3	The delivery does not satisfy the business' needs					Frequent production settlements are the most important measure to reduce this uncertainty	

9. References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile Software Development Methods: Review and Analysis. *VTT publication*, 1-107.

Alahyari, H., Svensson, R. B., & Gorschek, T. (2016, December 14). A study of value in agile software development organizations. *Journal of Systems and Software, 125*, pp. 271-288. doi:https://doi.org/10.1016/j.jss.2016.12.007

Anderson, C. (2010). Presenting and Evaluating Qualitative Research. *American Journal of Pharmaceutical Education, 74*, pp. 1-7. doi:https://doi.org/10.5688/aj7408141

Askheim, O. G., & Grenness, T. (2008). *Kvaliative metoder for markedsføring og organisasjonsfag.* Oslo: Universitetsforlaget.

Askheim, O., & Grenness, T. (2008). *Kvalitative metoder for markerdsføring og organisasjonsfag.* Oslo: Universitetsforlaget.

Auerbach, C. F., & Silverstein, L. B. (2003). *Qualitative Data: An Introduction to Coding and Analysis.* New York and London: New York University Press.

Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunnigham, W., Fowler, M., . . . Thomas, D. (2001). *Manifesto for Agile Software Development.* Retrieved from Manifesto for Agile Software Development: http://agilemanifesto.org/

Brevik, E., & Grønli, T.-M. (2013). *The Scrum Illusion: Use of Software Development Methods in the Norwegian IT-industry.* Tapir Akademisk Forlag: NOKOBIT2013. Retrieved from www.resear: https://www.researchgate.net/publication/259187914_The_Scrum_Illusion_Use_of_Software_Development_Methods_in_the_Norwegian_IT-industry

Burket, M., Ivens, B. S., & Shan, J. (2012). Governance mechanisms in domestic and international buyer–supplier relationships: An empirical study. *Industrial Marketing Management, 41*, pp. 544-556. doi:https://doi.org/10.1016/j.indmarman.2011.06.019

Carson, S., Mahok, A., & Wu, T. (2006). Uncertainty, Opportunism, and Governance: The Effects of Volatility and Ambiguity on Formal and Relational Contracting. *Academy of Management Journal, 49*, pp. 1058-1077. doi:https://doi.org/10.5465/amj.2006.22798187

Cobb, C. G. (2015). *Manager's Guide To Mastering Agile Principles and Practices for an Adaptive Approach.* New Jersey: John Wiley & Sons.

Cotteleer, M., & Sniderman, B. (2017, December 18). *www2.deloitte.com*. Retrieved from Insights: https://www2.deloitte.com/insights/us/en/focus/industry-4-0/overview.html

Den Norske Dataforeningen. (2018, June 1). *www.dataforeningen.no*. Retrieved from PS 2000 Smidig - Smidig Bilagssett:

https://www.dataforeningen.no/ps2000-smidig-smidig-bilagssett.4657604-147780.html

Denning, S. (2016, August 13). *www.forbes.com*. Retrieved from What Is Agile?: https://www.forbes.com/sites/stevedenning/2016/08/13/what-is-agile/#4707277226e3

Difi. (2018, February 7). *Direktoratet for forvaltning og IKT*. Retrieved from About Difi: https://www.difi.no/om-difi/about-difi

Difi. (2018, March 1). *www.anskaffelser.no, Difis fagsider of offentlige anskaffelser*. Retrieved from Statens Standardavtaler (SSA): https://www.anskaffelser.no/it/statens-standardavtaler/statens-standardavtaler-ssa

Dworkin, S. L. (2012, September 12). Sample Size Policy for Qualitative Studies Using In-Depth Interviews. *Archs. Sex Behav., 41*, pp. 1319-1320. doi:https://doi.org/10.1007/s10508-012-0016-6

Dönmez, D., & Grote, G. (2017). Two sides of the same coin - how agile software development teams approach uncertainty as threats and opportunities. *Information and Software Technology, 93*, pp. 94-111. doi:https://doi.org/10.1016/j.infsof.2017.08.015

ForbesInsights. (2018, January 22). *Forbes*. Retrieved from Forbes Insights with Deloitte: https://www.forbes.com/sites/insights-deloitte/2018/01/22/how-the-worlds-top-executives-are-approaching-the-fourth-industrial-revolution/#70dac967abd7

Gartner. (2017). *www.gartner.com*. Retrieved from Digitalization: https://www.gartner.com/it-glossary/digitalization/

Gartner. (2018). *Gartner*. Retrieved from IT Glossary: https://www.gartner.com/it-glossary/internet-of-things/

Ghosh, M., & John, G. (1999). Governance value analysis and marketing strategy. *The Journal of Marketing, 63*, pp. 131-145. doi:https://www.jstor.org/stable/1252107

Gibbons, R., & Henderson, R. (2012). Relational Contracts and Organizational Capabilities. *Organization Science, 32*, pp. 1350-1364. doi:https://doi.org/10.1287/orsc.1110.0715

Gilson, R. J., Sabel, C. F., & Scott, R. E. (2009). Contracting for Innovation: Vertical Disintegration and Interfirm Collaboration. *Columbia Law Review, 109*, pp. 431-502.

Graebner, M. E., Martin, J. A., & Roundy, P. T. (2012). Qualitative data: Cooking without a recipe. *Strategic Organization, 10*, pp. 276-284. doi:https://doi.org/10.1177/1476127012452821

Grønmo, S. (2010). *Samfunnsvitenskapelige metoder* (3 ed.). Bergen: Fagbokforlaget.

Haaskjold, E. (2013). *Kontraktsforpliktelser* (2. ed.). Cappelen Damm Akademisk.

Halvorsen, K. (2011). *Å forske på samfunnet, en innføring i samfunnsvitenskapelig metode* (5. utgave ed.). Oslo: Cappelesen Forlag AS.

Halvorsen, K. (2011). *Å forske på samfunnet, en innføring i samfunnsvitenskapelig metode* (5 ed.). Oslo: Cappelens Forlag AS.

Harash, A. (2016, November 8). *Huffington Post*. Retrieved from The Digital Revolution and its Impact on Industry, Consumers, and Government: https://www.huffingtonpost.com/entry/the-digital-revolution-and-its-impact-on-industry-consumers_us_57acdc9de4b0ae60ff020c2d

Heide, J. B., & John, G. (1990). Alliances in Industrial Purchasing: The Determinants of Joint Action in Buyer-Supplier Relationships. *Journal of Marketing Research, 27*, pp. 24-36. doi:https://www.jstor.org/stable/3172548

Heide, J. B., & John, G. (1992). Do Norms Matter in Marketing Relationships? *Journal of Marketing, 56*, pp. 32-44. doi:https://www.jstor.org/stable/1252040

Hendrikse, G. (2003). Chapter 5: Principal agent models. In G. Henrikse, *Economics and Management of Organizations* (pp. 90-98). McGraw-Hill.

Hendrikse, G. (2003). Chapter 6 Hidden action problem. In G. Henrikse, *Economics and Management of Organizations* (pp. 106-109, 112-122, 124-125, 129-131, 135-139, 145-147, 149-150, 155-156). McGraw-Hill.

Holton, G. A. (2004). Defining Risk. *Financial Analysts Journal, 60*, pp. 19-25. Retrieved from https://www.jstor.org/stable/4480615

Jiang, X., Gao, S., Bao, Y., & Jiang, F. (2012). Managing knowledge leakage in strategic alliances: The effects of trust and formal contracts. *Industrial Marketing Management, 42*, pp. 983-991. doi:https://doi.org/10.1016/j.indmarman.2013.03.013

Johannessen, A., Christoffersen, L., & Tufte, P. A. (2008). *Introduksjon til samfunnsvitenskapelig metode.* Oslo: Abstrakt Forlag AS.

Johannessen, A., Christoffersen, L., & Tufte, P. A. (2011). *Forskningsmetode for Økonomisk-Administrative fag* (3 ed.). Oslo: Abstrakt Forlag.

Jonnalagadda, G., Shafey, S., Lynah, W., Massetti, M., Maerten, P., Wieczorek, S., . . . Bonser, M. (2017, July). *PwC.* Retrieved from Agile Project Delivery Confidence Mitigate project risks and deliver value to your business: https://www.pwc.com/gx/en/actuarial-insurance-services/assets/agile-project-delivery-confidence.pdf

Jørgenrud, M. (2016, Jan 20). *www.digi.no*. Retrieved from NAV IT: https://www.digi.no/artikler/nav-bygger-helt-ny-it-avdeling/320700

Jørgensen, M. (2015). *Suksess og fiasko i offentlige IKT-prosjekter: En oppsummering av forskningsbasert kunnskap og evidensbaserte tiltak.* Oslo: Simula Research Laboratory, University in Oslo, Scienta.

King, N., & Horrocks, C. (2010). *Interviews in Qualitative Research.* SAGE.

Koch, A. S. (2005). *Agile Software Development Evaluating the Methods for Your Organization.* Boston: Artech House.

Kommunal-og moderniseringsdepartementet. (2015, June 16). *www.regjeringen.no*. Retrieved from Ønsker mindre IKT prosjekter: https://www.regjeringen.no/no/aktuelt/onsker-mindre-ikt-prosjekter/id2421621/

Kvale, S., & Brinkmann, S. (2009). *Det kvalitative forskningsintervju* (2 ed.). Oslo: Gyldendal Akademiske.

Kvale, S., & Brinkmann, S. (2009). *Det kvalitative forskningsintrvju* (2 ed.). Oslo: Gyldendal Akademiske.

Larman, C. (2004). *Agile and Iterative Development. A Managers Guide.* Boston: Addison-Wesley .

Llewllyn, K. N. (1931). What Price Contracts? An Essay in Perspective. (40), pp. 704-751.

Macaulay, S. (1963). Non-Conctratual Relations in Business. *American Sociological Review*, pp. 55-70.

Mannaro, K., Melis, M., & Marchesi, M. (2004). *Empirical Analysis on the Satisfaction of IT Employees Comparing XP Practices with Other Software Development Methodologies.* Eckstein J., Baumeister H. (eds) Extreme Programming and Agile Processes in Software Engineering. XP 2004. Lecture Notes in Computer Science, vol 3092, Computer Science. Cagliari: DIEE, Department of Electric and Electrical Engineering, University of Cagliari.

Marschall, T., & Korstvedt, T. (2017). *Digital Transformation Report 2017.* Oslo: QVARTS, Microsoft .

Miller, G. G. (2001). The Characteristics of Agile Software Processes. *Computer Society*, 1-3.

Moniruzzaman, A. B., & Hossain, S. (2013). Comparative Study on Agile software development methodologies. *Global Journal of Computer Science and Technology, 8*, pp. 5-18. Retrieved from https://www.researchgate.net/publication/249011841_Comparative_Study_on_Agile_software_development_methodologies

Myers, M. D. (2009). *Qualitative Research in Business & Management.* London, California, New Delhi, Singapore: Sage .

Palgrave . (2018). *Palgrave Study Skills*. Retrieved from Choosing appropriate research methodologies:

https://www.macmillanihe.com/studentstudyskills/page/choosing-appropriate-research-methodologies/

Patanakul, P. (2014). Managing large-scale IS/IT projects in the public sector: Problems and causes leading to poor performance. *Journal of High Technology Managment Research, 25*, pp. 21-35. doi:https://doi.org/10.1016/j.hitech.2013.12.004

Patton, M. Q. (1987). *How to Use Qualitative Methods in Evaluation.* Newbury Park, California: SAGE.

Poppo, L., & Zenger, T. (2002). Do Formal Contracts and Relational Governance Function as Substitutes or Complements? *Strategic Management Journal, 23*, pp. 707-725. doi:https://doi.org/10.1002/smj.249

Postholm, M. B. (2005). *Kvalitativ metode. En innføring med fokus på fenomenologi, etnografi og kasusstudier.* Oslo: Universitetsforlaget.

Pries, K. H., & Quigley, J. M. (2010). *Scrum Project Management* (Vol. 1). Boca Raton: CRC Press.

Procurement For All. (2018). *Procurement For All*. Retrieved from What is a Dynamic Purchase System?: https://www.procurementforall.co.uk/performance_measurement/what_is_a_dynamic_pu/

Regjeringen. (2013). *Meld. St. 23 (2012–2013) Digital agenda for Norge— IKT for vekst og verdiskaping.* Regjeringen. Retrieved from https://www.regjeringen.no/no/dokumenter/meld-st-23-20122013/id718084/

Richards, L. (2015). *Handling Qualitative Data A Practical Guide* (3 ed.). SAGE.

Rindfleisch, A., & Heide, J. B. (1997). Transaction Cost Analysis: Past, Present and Future Applications. *Journal of Marketing, 61*, pp. 30-54. doi:https://www.jstor.org/stable/1252085

Riordan, M. H., & Williamson, O. E. (1985). Asset specificity and economic organization. *International Journal of Industrial Organization, 3*, pp. 356-378. doi:https://doi.org/10.1016/0167-7187(85)90030-X

Ryen, A. (2002). *Det kvalitative intervjuet. Fra vitenskapsteori til feltarbeid. Kapitell 6: Det naturalistiske intervjuet.* Bergen: Fagbokforlaget.

Sande, J. B., & Haugland, S. A. (2015). Strategic Performance Effects of Misaligned Formal Contracting: The Mediating Role og Relational Contracting. *International Journal of Research in Marketing, 32*, pp. 187-194. doi:https://doi.org/10.1016/j.ijresmar.2015.02.002

Schepker, D. J., Won-Yong, O., Martynov, A., & Poppo, L. (2014). The many Futures of Contract: Moving Beyond Structure and Safeguarding to Coordination and Adaption". *Journal of Management, 40*, pp. 193-225. doi:https://doi.org/10.1177/0149206313491289

Silverman, D. (2005). *Doing Qualitative Research* (2 ed.). London: SAGE Publications.

Spence, I., & Bittner, K. (2005, May 15). *IBM developerWorks*. Retrieved from What is iterative development?: https://www.ibm.com/developerworks/rational/library/may05/bittner-spence/

Steendam, V. E., & Bergh, V. d. (2012). *Measuring Writing: Recent Insights into Theory, Methodology and Practice* (1 ed.). Boston: BRiLL.

The Norwegian National Committees for Research Ethics. (2014). *The Norwegian National Committees for Research Ethics*. Retrieved from Genereal guidelines for research : https://www.etikkom.no/en/ethical-guidelines-for-research/general-guidelines-for-research-ethics/

Tversky, A., & Fox, C. R. (1995). Weighing risk and uncertainty. *Psychological Review, 102*, pp. 269-283.

Williams, M. (2000). Interpretivism and Generalisation. *Sociology*, 209-224.

Williamson, O. (1979). Transaction-cost economics: the governance of contractual relations. *The Journal Law and Economics, 2*, pp. 233-261. doi:https://doi.org/10.1086/466942

Williamson, O. E. (1981). The Economics of Organization: The Transaction Cost Approach. *American Journal of Sociology, 87*, pp. 548-577. doi:https://doi.org/10.1086/227496

Yang, Q., Lu, T., Yao, T., & Zhang, B. (2014, July 5). The Impact of Uncertainty and Ambiguity related to iteration and overlapping on schedule of product development projects. *International Journal of Project Management*, pp. 827-837.

Østvang, T. (2016, Nov 15). *www.cw.no*. Retrieved from Karriere: http://www.cw.no/artikkel/karriere/nav-med-heftig-it-rekruttering