Thomas Nymo Skogvold

Agile Methods and Military Software Development

How do the Norwegian Armed Forces and agile vendors approach methodologies for software development?

Master thesis in Technology Management

Trondheim, October 2014

Supervisor: Dr. Arve Pettersen

Norwegian University of Science and Technology
Faculty of Social Science and Technology Management
Department of Industrial Economics and Technology Management

NTNU – Trondheim
Norwegian University of
Science and Technology

Abstract

Large governmental software projects have a dubious track record and numerous of terrifying projects have taken place within the military domain. The software industry is compared to other industries still in its infancy and numerous methodologies have endeavored to solve the inherent uncertainty in the software industry. Agile methods have achieved a lot of attention during the last decades and while some mention it as the new industry standard for software development others are skeptical and perceive the methods as a temporary hype. The Norwegian Armed Forces have accomplished some agile software projects. The heavyweight framework for military project management has still been applied, but vendors have been allowed to use their agile development processes. This thesis seeks to investigate this form of cooperation between the military agency for software acquisition and an agile vendor and the thesis' problem formulation is "How do the Norwegian Armed Forces and agile vendors approach methodologies for software development". The thesis starts with a literature review, but while the amounts of literature on agile development are overwhelming, most of it is of a normative character and written by authors who hold a polarized view and who have both feet planted in the agile camp. Reliable research on the methods is yet to be presented. Thus, the literature review is based on the discussions presented in recognized computer journals (*IEEE, Computer and Cutter IT*) and in addition "Balancing agility and discipline" by Boehm and Turner (2003) have to be mentioned as an important source of theory. The units of analysis are the *NDLO ICT* which is the agency within the Norwegian Armed Forces which deals with software acquisition and *Teleplan Globe,* a small actor in the Norwegian defense industry who has implemented agile development. The research suggests that it can be beneficial for the Norwegian Armed Forces to employ agile vendors as the methods seem to have certain benefits compared to traditional development. Agile development involves few personnel and can be cost effective and seems to have an advantage regarding close interaction between users and developers. However, agile methods can be problematic when life-critical systems are developed, when thorough documentation is essential or when a large group of developers have to cooperate.

<u>Keywords:</u>

Military software development, agile software development, Norwegian Armed Forces project management, Prinsix, Norwegian Defense Logistics Organization, NDLO ICT, Teleplan Globe, agile vendors, agile contractors, governmental software acquisition.

Acknowledgements

The author would like to thank the informants who participated and shared their stories of problems and achievements. Despite hectic work days filled to the brim with meetings and deliverables, you were able to reserve time in for interviews and conversations. This thesis would not have been possible without your contribution.

The author would like to give special thanks to Arve Pettersen, the academic supervisor who mentored the author through the process. Your honesty and straightforwardness has been a source of gratification and frustration which both have been greatly appreciated.

The author would also like to thank friends and family who have been turned down too many times during the last months. You have either understood my situation or given me up already – time will show. After turning down one suggested activity after the other, a friend remarked that this thesis was my *Sagrada Família* – a project neither of us would live to see completed. The last sentence is now written and the final dot is in its place. This is an emotional moment, of joy and melancholy. It marks the end of an instructive journey and from here the voyage continues into uncharted waters. However, good things lie ahead, and without comparison, it looks like even the perpetual Catalonian project might be admired in complete form already in 2028.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CMM	Compatibility Maturity Model
DoD	(United States) Department of Defense
FACNAV	Forward Air Control and Navigation (a mapping software)
FFI	Norwegian Defense Research Establishment (Forsvarets forskningsinstitutt)
ICT	Information and Communications Technology
IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technology
NATO	North Atlantic Treaty Organization
NcW	Network-centric Warfare
NDAA	National Defense Authorization Act
NDLO	Norwegian Defense Logistics Organization
NORBMS	Norwegian Battle Management System (further development of FACNAV)
OPS	Operations
POC	Point Of Contact
QA	Quality Assurance
RMA	Revolution in Military Affairs
RUP	Rationalized Unified Process
STD	Standard
US	United States

1 Introduction

The cold war ended with the fall of the Soviet Union in 1991, and it was seen as a triumph to western military high-tech. Operation Desert Storm against Iraq was commenced the same year, and the swift victory was thought to represent a historical shift in the geopolitical situation *–the Revolution in Military Affairs* (RMA). Later experience shows that enemies are increasingly able to pick battles where western technology is irrelevant and soldiers, military think tanks and politicians inquire the effects of the revolution. The RMA is today usually used as a general term explaining modern information technology's (IT) impact on warfare. Today the RMA is rarely mentioned and the decisive triumph seems long-forgotten.

In 2003 the US Department of Defense (DoD) issued the directive *Joint Vision*, which initiated a new trend regarding investments in technology and experiments. The goal was to achieve more flexibility when fighting the modern war, by shifting from a platform centric to a network centric military. One impediment lied and lies in getting thousands of proprietary military software systems to communicate in safe ways using shared standards. The US has traditionally guided global military development, and western militaries are currently working side-by-side facing the challenges of implementing Network centric Warfare (NcW). The goal is to use modern IT to achieve increased speed of command and control and to allow self-synchronization between units. Safe communication is essential, especially when task forces increasingly often are from different nations (combined ops) and branches (joint ops) and under unified command (Daltveit, Ydstebø, & Geiner, 2010).

The military organizations are usually perceived as highly bureaucratic and strictly hierarchical. Chain of command, centralized decisions, task specialization, discipline and standard operating procedures are common terms from the daily routine for military personnel. Morgan (2006) uses the military organizations as examples when he describes organizations where the machine metaphor is appropriate. These organizations have their strengths in stable environments, -when tasks can be standardized and when precision is vital. The downside is tendencies to foster unwieldy bureaucracies that are poorly adjusted to changing environments (Morgan, 2006) and the military acquisition systems *are* known as costly and financially inefficient. Since the beginning of software development, software projects have been feared because of their inherent uncertainty and risk (Alleman, 2002). A 1995 report on DoD software projects exposed that 75% of software projects failed and only 2% of the completed projects could be put to use without massive modification (Larman, 2004). Although this is a depressive track record, it is the offspring of the same acquisition system which played a vital role in winning the cold war's contest of weapons development (Sapolsky, Gholz, & Kaufman, 1999). In other words, the system has its success stories, but it is hard to argue against that there is a great underlying potential for improvement, and perhaps especially in the field of software development. Brooks Jr (1987) compared software projects with warewolves, because of their frightening habbit of turning from the familiar into the worst horrors. He claimed that there was no silver bullet in sight, –neither in management science nor in technology, that could put these monsters to rest.

Advocates of agile methodologies have long claimed that their methods are a panacea against the traditional problems of software development. However, just like the famous axiom in medicine which tells that all effective medications have their side effects, the agility is likely to come with a cost, and that the methods should probably be applied wisely and with care. The Norwegian Armed Forces has some experience with agile methods through vendors. The FACNAV project which started in 2006 is regarded as the first agile project and it was declared an unusual success; it was "cheap" (compared to similar projects), it was user friendly, it was of high quality and it was also delivered on time (Danielsen & Valaker, 2012). This stands in great contrast to the usual newspaper stories which tell tales of exceeded budgets, late deliveries and products of questionable quality. Can agile methodology be the long sought silver bullet?

1.1 Background

Military budgets are increasingly subject to scrutiny and we need to utilize every dollar of funding. Winston Churchill is said to have expressed "Gentlemen, we have run out of money; now we have to think" (unsourced quote) and smarter ways to organize projects are likely to have a significant impact on the cost-effectiveness of military organizations. The US DoD is globally one of the most important procurement organizations, and they have been at the frontline searching for smarter and more cost-effective methods, perhaps especially in the field of software development. Since the early days of modern IT the organization has more or less dictated the standards for software development. NATO has in large portions adapted the same standards, and Norway is no exception. Compliance with these standards has been prerequisites for vendors competing for governmental contracts and governmental organizations, and private enterprises have followed in the wake and adopted similar standards. These standards have changed dramatically since the early days of software engineering. From demanding a linear waterfall practice the DoD have gradually opened up for iterative and incremental development and are currently making way for agile practices. Even the most successful machines need regular maintenance and modernization. Is the Norwegian military acquisition machinery in need of some drops of oil or a comprehensive reengineering when it comes to acquiring software?

1.2 Problem formulation and research questions

This thesis seeks to cast light on software development methodologies in projects initiated by the Norwegian Armed Forces. It investigates the status within the Norwegian Armed Forces' department for ICT-procurement and within one of their vendors who practices agile development. The problem formulation is *How do the Norwegian Armed Forces and agile vendors approach methodologies for software development*?

The research questions are written based upon the problem formulation and the literature review presented in chapter 2:

1. What are the characteristics of the methodology applied by the Norwegian Armed Forces and of the methodologies applied by their agile vendors?
2. To what extent are projects pre-planned? Are plans followed mechanically or used as a basis for continuous modifications as projects progress?
3. What are the experienced challenges and critical success factors with the applied methodologies?

1.3 Limitations of the study

The reviewed litterature on methodology origins from a paradigm which expects a casual connection between actions of management and outcomes of projects. In real life, these connections are apparently more complex, and although methodologies can play important roles, it is assumed that no methodology can guarantee succesfull software development. – No methodology alone is likely to make up for the value of factors like personell, leadership, education or experience. However, the thesis discusses methodologies with a basis in this paradigm.

The thesis is written as an unclassified educational thesis, therefore, material which the researcher gained access to or was presented during the data gathering process is not included in the final report. One report which was used by the researcher was labeled "Exempted from public disclosure" and this report is referred to in chapter 4.1.4. The lead researcher behind the report granted the author permission to reproduce some of its conclusions in this thesis.

Agile is a wide term and several methodologies are said to fall under the agile umbrella. However, the thesis will often mention agile as if it was a concrete methodology. No attempts will be made to contrast and compare specific agile methodologies, but a brief overview of the Scrum methodology is presented in chapter 2.4.3.

1.4 Structure of the thesis

Chapter 1: Introduction: This section introduces the study, presents the problem formulation, the research questions and the limitations of the study.

Chapter 2: Literature review: Starts with an introduction of project management terms derived from pilot interviews and early literature studies. The iron triangle, linear, incremental and iterative practices will be explained as they are believed to be relevant for the thesis. Then the linear waterfall model is explained, before "real-life" waterfall is presented which can actually be both iterative and incremental. Then the thesis presents the agile methods and discusses various definitions, before an agile methodology (Scrum) is briefly presented. The US Department of Defense has had a huge impact on software development methodology and the most important standards are mentioned, before a potential breakthrough for agile methods in (US) military software acquisition is briefly introduced. The agile methods can either be a temporary hype or an important step towards better managed software projects and some views from the ongoing discussion is presented, before the literature review is concluded with a brief summary and research questions are presented.

Chapter 3: Methodology: This section starts with a brief introduction of research paradigms which leads further to research strategies and the applied research design which is case study research. The concrete methods for data gathering are described under data collection, before different data sampling approaches are discussed. Then the quality of research and ethical considerations are discussed.

Chapter 4: Empirical Results: This section lists the empirical data from the two cases. They are presented individually using a framework which is drawn from the research questions.

Chapter 5: Cross case analyses: The empirical results have already been presented individually and this section merges the highlights from the data material and tries to contrast, and to compare.

Chapter 6: Recommendations and Conclusions: This section concludes the study by summarizing the most important findings before it suggests measures for improvement and proposes areas for future research.

2 Literature review

The ensuing litterature compilation is an attempt to reproduce the ongoing discussion and provide an unbiased background for the thesis research objectives. The main theoretical bases were identified as Project Management Methodologies, Linear and Sequential Development, the Waterfall model, Plan Driven development, Software Development Methodology, Iterative and Incremental Development, Agile Software Development and Department of Defense (DoD) Software Acquisition. These bases were used as keywords in different combinations in different gateways such as Bibsys, WorldCat and especially Google Scholar. This searching strategy proved valuable when searching for material in relevant journals. Computer and software journals such as Computer, IEEE Software, IEEE Computer and the Cutter IT journal were excellent sources of information as they had already hosted a discussion on agile vs traditional methods for more than a decade. Several non-computer related journals were also helpful, as the same discussion has diffused from software development to many other industries. Journal articles were helpful to understand the topic, but their tables of references were also of great value. These tables were used to find more material on the topic; journal articles, books, essays, white-papers and webpages. While there is vast amounts of literature and research available on the traditional methods, the agile methods are still in their infancy and while normative litterature written by practitioners or consultants, academic research is still scarce. It should be mentioned that agile methods are not an "opposite" of the traditional methods, but in order to answer the thesis question a polarized view is maintained; differences are highlighted rather than their similarities.

2.1 Terminology in Project Management derived from pilot interviews and early literature studies

The first round of interviews revealed that some terms were frequently brought up for discussion. The price-cost-quality triangle, linear, incremental, iterative and sequential development was identified as especially relevant for the thesis and is therefore presented in the succeeding subchapters.

2.1.1 The triple constraints or "the iron triangle"

Project managers are constantly facing the challenge of the triple constraints or the iron triangle; delivering on time, to budget and with the required quality. The factors are interrelated and you cannot alter one factor without at least affecting one of the other

(Atkinson, 1999, illustrated in Figure 2-1 The iron triangle). You can deliver good quality in time, but it will be costly, you can deliver to cost and at time, but product quality will hurt and you can deliver to cost with desired quality, but the project will be delayed (Wysocki, 2009). The challenge appears to be universal, across industries and cultures and little indicates that military project organizations are any different.

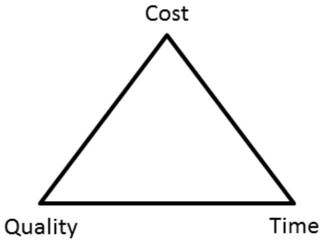

Figure 2-1 The iron triangle

Source: Atkinson (1999)

2.1.2 Linear development

Linear development means that a progress can be thought of as a straight line from start to finish, with no steps backward underway. A thought project consists of *n* identified steps (or milestones). If these steps are addressed and completed in their preplanned sequence the project is linear.

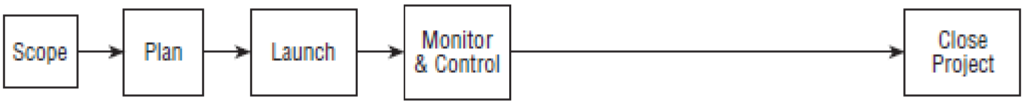

Figure 2-2 Linear project management

Source: Wysocki (2009)

The figure shows how the stages are executed successively and note that there is no looping back in linear development, which would allow learning from one phase to revise previous work. E.g. if a project team gains new knowledge during the launch phase, they cannot alter the plan if they follow a strict linear model. The linear model works when requirements can be defined early and when no significant change is expected during the development. Kruchten (2001) explains that a common problem with linear models is that the majority of rework comes near the end of a project as these projects often discovers bugs and problems during the testing phase and huge amounts of rework can set linear projects back to zero (Royce, 1970).

2.1.3 Incremental development – "add onto what you have"

Larman (2004, p. 20) explains incremental as "a practice of delivering a system in a series of expanding capabilities" and the series of releases are usually referred to as increments. Incremental development allows early business value, as components are added to a system as they are completed. This allows users to interact early with prototypes, which can be very valuable for developers, and Boehm (1996) claims that a prototype might exceed the value of 100,000 words of written documentation. Early deliveries reduce the chance of developers losing sync with the users' actual needs, which usually leads to excessive rework. An axiom in the modern IT-industry is that great "great software is not built, it's grown" (Monson-Haefel, 2009, p. 194) and it can be very beneficial to let users interact with early prototypes instead of almost complete systems (Cockburn, 2008). Figure 2-3 shows an incremental model, where expanding capabilities are delivered with each increment. Note that increments do not have any looping back to revise plans. The benefits of incremental development and customer interaction are further explained in chapter 2.5.3 - User involvement.

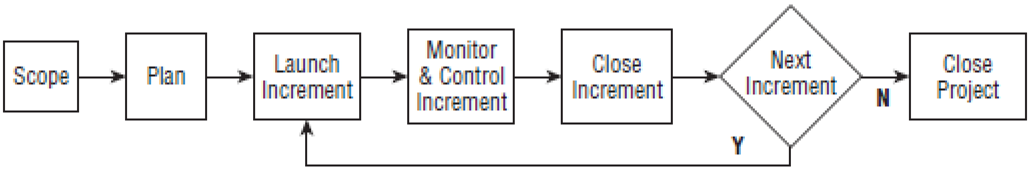

Figure 2-3 Incremental project management

Source: Wysocki (2009)

2.1.4 Iterative development – "Stop, evaluate and rework"

Iterative development means to have a plan and evaluate the product at certain points during the development. If the software is tested and minor problems are discovered the developers can correct these problems and continue, but any major problems would cause a major setback followed by massive rework (Cockburn, 2008). An iteration is the result of repeating a sequence of operations with the aim of getting closer to the desired outcome. Each iteration is meant to result in an iteration release, a partially complete system. While some iterations are presented to the customers, most iteration releases are internal as they are meant for evaluation by the developers (Larman, 2004), but it is important to maintain continuous feedback from customer representatives (Cockburn, 2008). Figure 2-4 Iterative project management lifecycle model shows how iterative development lets the team use learning in later stages to revise the plan (and if necessary the requirements which is the foundation for the plan).

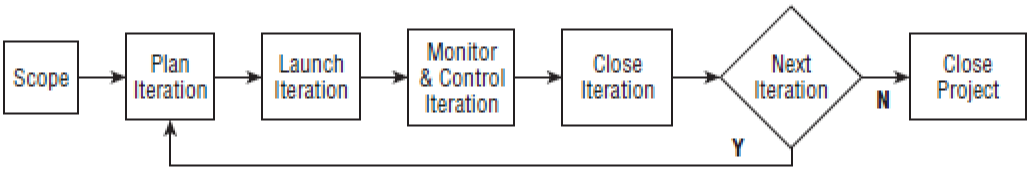

Figure 2-4 Iterative project management lifecycle model

Source: Wysocki (2009)

If necessary, requirements can be rewritten and design might be altered until final testing results in deployment of software. The recommended length of these iterations varies, but between one and six weeks is generally recommended (Larman, 2004, p. 11). Iterative development does not exclusively have positive effects; projects do not get easier to set up, to plan or to control because they are iterative. A project manager will face greater challenges with iterative development, especially during early iterations when risks are high and early failure possible. However, iterative development has a tendency to simplify the work for developers (Kruchten, 2001; Ruparelia, 2010). Iterative development is beneficial when requirements are not complete or are likely to change as a project progresses, or when some parts of the solution or parts of functionality is not identified when a development starts.

In project management these steps (or milestones) are usually categorized in stages. In software development these stages are typically: requirements, analysis, coding, testing and delivery. Stages characterize *sequential models* and since the described models are sequential, they could be referred to as linear-sequential, incremental-sequential and iterative-sequential, but since all models referred to in this thesis are sequential, the models are referred to as linear, incremental or iterative.

2.2 The Waterfall model

Project management has traditionally been a linear process where the completion of a stage has led directly to initiation of the succeeding stage. The model is commonly referred to as “the waterfall model” and the name derives from the common graphic illustration, showing stages as steps in a waterfall (see Figure 2-5). Just as the laws of physics does not allow “water” to flow back up in a waterfall, a completed stage in a waterfall project initiates the succeeding stage, without any looping back. In the IT sector the waterfall model usually starts with requirements, and when they are negotiated, a project plan is written, which describes the subsequent phases: design, development, testing and delivery. When the plan is completed the focus shifts from planning to execution. (Ruparelia, 2010) emphasizes that this is a descriptive model and not a methodology in the sense that while a model describes what to do a methodology also describes how to do it.

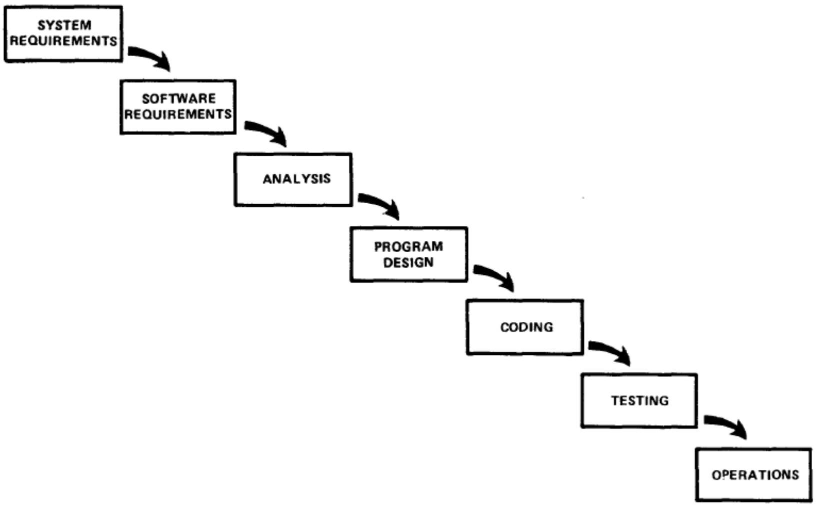

Figure 2-5 The waterfall model

Source: Royce (1970)

The figure above illustrates the one way stream of the waterfall model. A problem is that this is often revealed in the late stages. Boehm (2006) has studied how the cost of change increases with the level of completion in a project. His research has led to the conclusion that the cost of change increases exponentially with the level of completion (see Figure 2-6 below). Discovering and dealing with mistakes or implementing changes should be done as early as possible in the development.

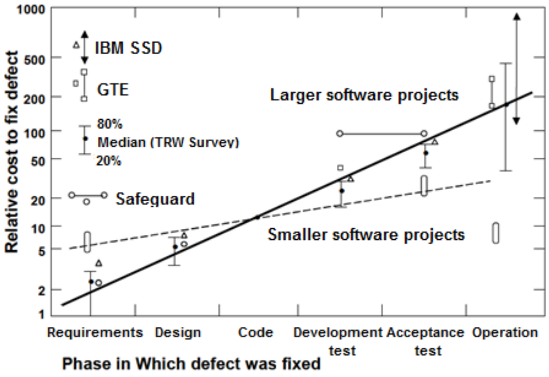

Figure 2-6 Cost to fix vs phase

Source: Boehm (2006)

The rationale behind the waterfall model derives from the industrial revolution, where work was standardized and highly repeatable. This ensured that workers were easy to replace, individuals could move from project to project without time-consuming relearning (Buresh, 2008) and it gave managers an impression of control as it was easy to measure progress with stages and milestones. The waterfall model was first described in 1956 by Benington, but an article by Royce (1970) is often mentioned as the origin of the model.

2.3 Real-life Waterfall and Plan-driven development

One of the main questions with the waterfall model is whether software projects can actually follow a prefixed plan or if the inherent nature of software development is too unpredictable. While the waterfall model works well in hardware development, software is different in its nature (Buresh, 2008). According to Boehm and Turner (2003) the difference between hardware and software development is that the former faces physical constraints while the latter deals with logical constraints. Physical constraints can often be predictive, but this is not necessarily the case with logical constraints such as the constraints in software development projects.

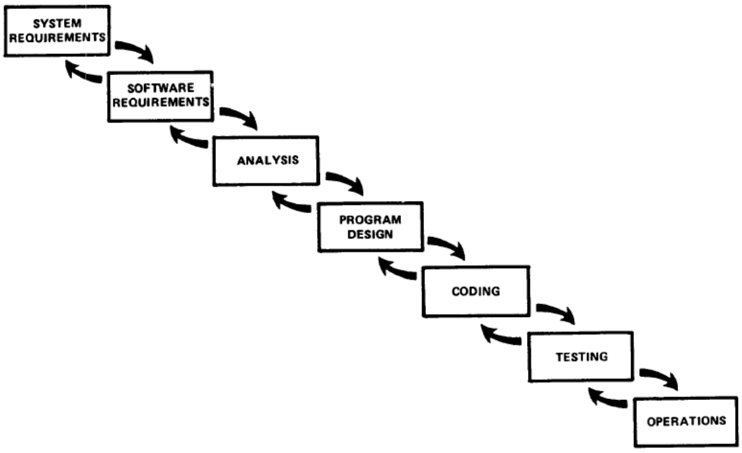

Figure 2-7 “Real-life Waterfall”

Source: Royce (1970)

The figure above has a lot in common with the waterfall figures that are often referred to by agile “fundamentalists” (similar to Figure 2-5), except that this figure has arrows drawn in both directions. This indicates a less rigid methodology than the single pass waterfall. Experience and learning during the development phase can lead to rework in previous stages. In theory, this Waterfall model can actually be iterative if implemented correctly. Alleman

(2002, p. 5) claims that the term waterfall has been used unfairly as a “strawman” by the agile community and that few, if any, pure waterfall projects exists today (at least not in the software industry); change and learning loops have been added to the model (iterative development) which opens up for evaluation looping back in the development process when previous work needs to be revised.

Boehm and Turner (2003) use the term “plan-driven” methods instead of waterfall, probably since the term waterfall has a negative tone to it in the modern IT industry. The plan-driven methods are said to have an important advantage in their potential for reuse of existing software that are stored in component libraries and a plan-driven staff is usually trained in writing documentation effectively (Buresh, 2008). Learning is essential for plan-driven methods, and is often referred to as process improvement. Specific activities are designed to improve the performance and the maturity of an organization’s processes, or simply ensure learning (Highsmith 2004, forwarded by Buresh, 2008). The Spiral Model, Rational Unified Process, and the Compatibility Maturity Models (CMM –can be argued to be a framework rather than a methodology) are some examples of modern plan driven models.

2.4 Agile development

The Merriam-Webster Dictionary explains the word agile as: “having a quick resourceful and adaptable character” and having an “ability to move with quick easy grace”. In software development “agile” is a collective term for lightweight methodologies, where lightweight refers the total weight of “control elements, including deliverables, standards, activities, milestones, quality measures and so on” Cockburn (2000, p. 65). The term “agile” was coined when a group of practitioners of different lightweight methodologies met in 2001 and they founded the self-named “Agile Alliance” which issued “the Agile Manifesto”.

2.4.1 The Agile Manifesto

Seventeen practitioners of lightweight software methodologies met during three days in 2001 to ski, relax and discuss software development. They were surprised to learn how much they had in common and they were able to agree to a common set of values for good software development. Their focus was on trust and respect of individuals rather that processes and procedures. The results of the meeting was written down and signed by all participants, and the result was named “the Manifesto for Agile Software Development” or more commonly referred to as *the Agile Manifesto* (Highsmith, 2001). The manifesto defines four values that are backed up by twelve principles (see Appendix A – The Agile Manifesto) which together constitute the core of the agile movement:

> We are uncovering better ways of developing software by doing it and helping others doing it. Through this work we have come to value: Individuals and interactions over process and tools, Working software over comprehensive documentation, Customer collaboration over contract negotiation, Responding to change over following a plan. That is, where there is value in the items on the right, we value the items on the left more (Beck et al., 2001).

2.4.2 Definitions of Agile Development

Although the quote above indicates what agile practitioners should value, and the manifesto describes agile as "early delivery of business value" (Beck et al., 2001) it is nowhere near a precise definition. There is no general, unanimous definition of the agile methodologies and therefore the following table is included. It is meant to elaborate the text from the manifesto.

Ambler:
[Agile is] "…an iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework with "just enough" ceremony that produces high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders" Source: Ambler (2005)
Cockburn:
"Agile is early delivery of business value. That involves early and regular delivery of working software, a focus on team communications, and close interaction with users". Source: http://bradapp.blogspot.com/2006/05/nutshell-definitions-of-agile.html
Boehm and Turner:
[The agile methods are] "lightweight processes that employ short iterative cycles; actively involve users to establish, prioritize, and verify requirements; and rely on tacit knowledge within a team as opposed to documentation" Source: Boehm and Turner (2003, p. 17)

Table 2-1 Definitions of Agile Development

These attempted definitions reveal that agile is iterative development, (development that is open to changes) and development done in close collaboration with users. Incremental development is indicated, and expressed explicitly in Ambler's (2005) definition. Boehm and

Turner's (2003) definition states that agile methods regard tacit knowledge as superior to documented knowledge, which is closely connected with the focus on team communication emphasized in Cockburn's definition.

2.4.3 An example of an agile method -Scrum

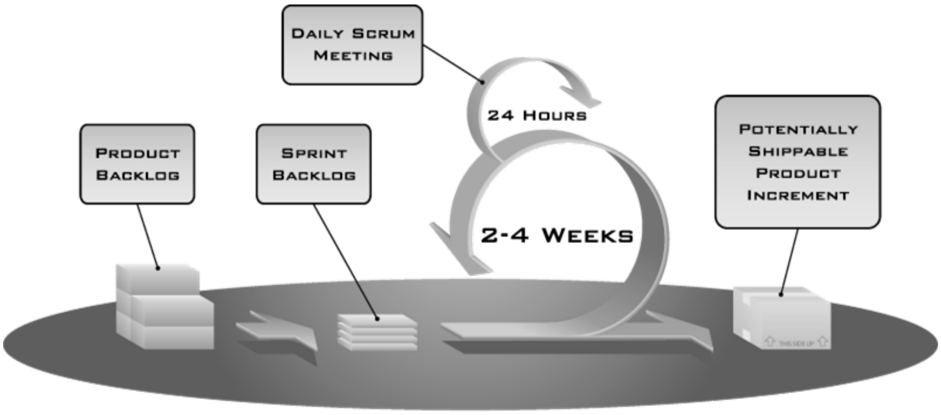

Figure 2-8 Scrum, combining iterative and incremental development

Source: http://www.mountaingoatsoftware.com

Figure 2-8 gives a simplified graphic representation of a Scrum process. The Product Backlog is a prioritized list of features that needs to be developed and the intention is that only the highest priority functions are implemented first. Before each Sprint (or iteration) the Product Owner (customer representative) picks out a set of features to be developed during the next sprint. This Sprint Backlog is negotiated with the team. The goal for a sprint is to complete the Sprint Backlog. The large green circle represents the sprint which is typically one month. During the Sprint, each working day starts with a Daily Standup Meeting (team members are standing to ensure the meeting is kept brief), where the team members provide an update on their work and inform others if they have encountered any impediments and what assistance is required (Schwaber, 2004).

2.5 Methodology home grounds

This section is especially inspired by Cockburn (2000) and his suggested factors to estimate whether a project is better suited with lightweight or heavyweight methodology and the description of "home-grounds" for plan-driven and agile methodologies by Boehm and Turner (2003). The authors agree that agile project management is not favorable for all types of organizations, projects or project teams and that the following considerations need to be considered before choosing the best suited methodology for a certain project. These factors are interrelated and several other classifications might be reasonable, however, the author found this classification useful to summarize differences between the two methodologies and potential impediments for successful application of them in certain types of projects. A summary is presented in Table 2-2 in section 2.5.8 at the end of this subchapter.

> It matters greatly whether the project sponsors want to have the software soon, want it defect free, or want to have the process visible (Cockburn, 2000, p. 68).

2.5.1 Requirements

Requirements need to reflect the various needs of different stakeholders; customers, users, purchasing agents, support personnel, maintenance etc. Their needs should be taken into consideration throughout projects, not only in the planning phase. Understanding the product life-cycle is essential; from the initiation of a project, through development, implementation, maintenance, and eventually out-phasing. Successful identification of requirements is essential irrespective of methodology and one of the main differences between the methods is how early this identification should and can be done. Traditional waterfall projects try to minimize the waste of work that is done by defining requirements early, before a plan is developed and the focus shifts from planning to execution and the practice does in theory not allow change in requirements during development. Modern plan-driven approaches acknowledges some degree of change in requirements during a project, but is best suited if requirements are relatively stable (Boehm & Turner, 2003). Kruchten (2001) argues that changing the underlying technology in a project in the later phases is a risky approach and should be avoided. The agile alliance tells practitioners to “welcome changing requirements, even late in development”, -it is actually their second principle (Beck et al., 2001). This raises a question whether software requirements are emergent or pre-specifiable. This will of course vary with the type of project, while a routine project like creating inventory software for a department might be pre-specifiable; an exploratory project on cyber warfare is likely to have a high degree of emerging requirements. Major software projects can be in development process for years, and if a project’s technology is limited to the technology that was available when the project plan was written, the customer might get an out-of-date product when the product is completed. DeBellis and Haapala (1995) refer to this phenomenon as a *technology freeze*. Similar problems occur in more tangible projects as well. The platforms that were developed during the cold war’s race in weapons development were created with requirements as the driver behind projects. However, such platforms are often results of decades with project work before even the first screw is put in its place. The Norwegian frigate project is one such example, it took more than two decades from the first requirements were written to the first platform was delivered. In the meantime there had been a revolution in several areas, e.g. electronic navigation equipment. This resulted in a need for massive upgrading of the platform even before it was delivered. The alternative would have been brand new frigates that were technically out-of-date.

2.5.2 Knowledge sharing – oral or written

The agile advocates claim that requirements are important, but no more documentation should be made than absolutely necessary. Boehm (2002) claims that this might work as long as the tacit knowledge is sufficient, but that it is a risky approach: if the tacit knowledge is insufficient, some life-cycle needs are easily overlooked. Paulk (2002) shares this concern, and also warns against relying too much in an unaided memory; reliance of tacit knowledge alone makes projects vulnerable to perception shifts over time. Buresh (2008) emphasizes that frequent evaluation of work is important for agile methods, as it encourages the teams to learn and reflect on previous practices, which leads to more accurate development in the future, and tacit knowledge is thereby gained by the team instead of being recorded in documents. Cockburn (2000) explains that learning and communication are closely related, and practices that encourage two-way communication is much more effective than practices that rely on huge amounts of shared documents.

The military organizations are often associated with strict standards, and these standards serve their purpose. Security standards do not give room for experimentation and creativity, but need to be followed strictly and documented elaborately. In the era of NcW, Norwegian forces are dependent on access to coalition networks, which might be an impediment for agile practices and their reliance in tacit knowledge. However, in projects where interoperability and security is not an issue, the author expects that agile projects can be very cost effective compared to traditional ways. Another issue is that the military organization has a high turnover rate, and tacit knowledge makes projects vulnerable to personnel changes.

2.5.3 User involvement

It seems self-evident that software projects need to understand the needs of their end-user in order to develop successful software, but while customers and users are probably most able to explain what they need, they are rarely able to specify in detail (Ambler, 2006). Even when requirements are well described in a document, users have a tendency to change their minds as soon as they interact with early versions of the software (DeBellis & Haapala, 1995). Boehm (1996) claims that letting customers interact with prototypes is much more valuable than documents. Customers might change their minds when they are presented with what they described. It is important to have a contract which defines rights and responsibilities, but having effective communication with the customers is even more important as customers and developers must educate each other during a project (Ambler, 2006). Agile developers want to work closely together with customer representatives that are dedicated to the development process, and when customers lack a good sense of direction, Highsmith and Cockburn (2001) stress that agile developers should follow them around and give customers reminders when necessary, since poor customers leads to poor software. A common problem is that the product team takes over the customer interaction responsibilities, and the product team starts thinking they understand customer needs, and make decisions on behalf of customers. This is also seen often in the military, staff officers with no recent field experience are often used to represent user interests in projects. This has resulted in several sub-optimal products, which emphasizes that it is preferable that users are represented at least by some representatives that are actual present users, rather than staff officers with antiquated operational experience. The way the Norwegian forces operate has changed dramatically during the last decade, but several products have been developed which made great sense ten years ago, but that are sub-optimal on the modern battlefield.

2.5.4 Plan or Improvisation

Paulk (2002) elaborates that continual planning and preparations *are* prerequisites for success, regardless of methodology, and the agile methods are no exception. This view is supported by Kruchten (2001) who expresses that agile development is *not* letting developers start coding and designing without plans and goals while hoping that clear goals and plan will emerge later on. Planning for change is very different from not planning at all. Boehm (2002) explains that the agile methods are in fact much more plan oriented than people think, because they stress the planning process more than the resulting documents. Developing a plan has its positive effects. It forces team members to think through the project and possible contingencies, but a plan is usually outdated within a few days, and the team should then focus on the changing realities rather than the outdated plan (Ambler, 2006; Highsmith & Cockburn, 2001).

There famous military axiom “Plan is nothing, planning is everything” does not mean that plans are worthless. A common way of operations planning is that a planning team presents a plan for senior staff officers. Every part of the plan is then challenged with “what if questions” and it is important that the plan is flexible enough to deal with contingencies. The process shares this objective of planning for change with the agile methods. The importance of the plan itself seems to increase with the number of people and elements involved. While it is relatively uncomplicated to improvise in smaller teams, it gets increasingly difficult with more people, platforms and support units involved. The planning process is paramount, but the plan itself is also of great importance.

2.5.5 Size

According to Cockburn (2000) any pause done during development to coordinate with other people or to update documents comes with a cost, and such pauses are more frequent in bigger teams. Agile development is difficult for large projects since real-time communication on a personal basis is difficult to achieve (Highsmith & Cockburn, 2001; Ruparelia, 2010). The average agile project has only nine people, although Cockburn and Highsmith (2001, p. 133) claim that successful agile projects have been found involving up to 250 people. However, Constantine (2001) expresses that the agile methodologies are difficult as soon as the team exceeds 15-20 developers. Cockburn (2000) points out that the need for control interrelates with the size of the development team, but as one of the original agile signers, he adds that greater problems does not necessarily demand larger development teams.

One of the benefits of smaller teams is their potential for effective communication. Ruparelia (2010) states this as an impediment for larger teams to apply agile practices, as agile emphasizes real-time communication, preferably on a face to face basis. This view is supported by Cockburn who expresses that the effectiveness of communication correlates negatively with the number of members on a team. While two people discussing a problem on a whiteboard is a very efficient way to communicate it is difficult to achieve the same levels of communication for large teams (Cockburn, 2000). Boehm (2002) argues that plan driven methods are better suited to manage larger projects, but that the heavyweight methods can be very inefficient for smaller projects.

2.5.6 Criticality

Cockburn (2000) divides the level of criticality in projects in four categories; loss of comfort, loss of irreplaceable money (will lead to bankruptcy), loss of discretionary money (common in exploratory project) and loss of life. He introduces a scorecard which is meant to estimate whether a project is fit for light weight (agile) or heavy weight (plan-driven) methods. He argues that high criticality calls for heavier methods, and this is supported by Ambler who expresses that he “would be leery of applying agile modeling to life-critical systems” (Ambler, 2001). Cockburn (2000) argues that the agile practices can be adjusted to work well with systems that are regarded as critical. Boehm and Turner (2003) argue that the agile methods are currently untested on safety-critical products and are concerned by the agile methods lack of documentation. They claim that plan-driven methods are preferable when products with great underlying criticality are developed. The Norwegian Armed Forces deal with events and products in both ends of the “Cockburn-scale” and although e.g. weapon guidance systems uncourtly are of great criticality (a few meters inaccuracy might lead to collateral damage or fratricide) there are also numerous systems in the other end of the spectrum. The author will argue that the organization needs methods that can handle development of software with great criticality, and the amount of documentation seems to be a key issue. However, it can probably be cost effective to apply lighter methods when products in the other end of the scale are developed.

2.5.7 Culture and personnel

In agile cultures, projects seem less reliant on plans, i.e. it is of great importance that the personnel involved in development have a good understanding of other developers’ work. This is critical to ensure successful integration. People who thrive under plan-driven cultures seem to prefer clear policies and procedures and have well-defined tasks that follow their position. The extensive project plan is there to ensure that individual contributions can be integrated, and it therefore limits the needed knowledge and understanding of the work of others. Boehm and Turner (2003) claim that people have a strong tendency to self-select their preferred culture and that agile methods are more likely to work in cultures that “thrive on chaos” while plan-driven methods work better where people “thrive on order”. Buresh (2008) claims that one of the problems with plan-driven approaches is that software developers detest documentation and sometimes ignore or even openly rebel against the need for documentation. The agile manifesto with its focus on people over process seems to reflect this view, and a lot of the flexibility of the agile methods is believed to reside in the absence of elaborate formal documentation. This increases the importance of tacit knowledge.

The Norwegian Armed Forces seem to fit best in the “thrive on order” category. Among the combat units, discipline and structure is essential. From the first day of service soldiers are pushed towards fitting into a defined frame. The author’s impression after fifteen years in the military is that the rigidness varies with branches, units and certainly individuals. The conventional army is often perceived to be very rigid while the Navy and Air Force are less dependent of structure. However, everything that regulates safety seems to be strictly standardized in all branches. Special Forces are often believed to be less dependent on structure than conventional counterparts and the project organization is believed to be very dependent on standards and process discipline. Personal impressions are always victim to biases, however the author believes that there is a core of substance in these prejudices.

Highsmith and Cockburn (2001) claim that a small group of designers can produce a better design than each could produce by alone; however Boehm and Turner (2003) question this statement and claim that the agile methods are dependent of a critical mass of premium people, and without premium people the agile methods are likely to result in inefficient projects. They emphasize that all methods work better with skilled people, but a well-defined plan makes it possible for less-capable people to contribute with low risk. In plan driven projects the most capable people typically work more during the early stages when the plan is developed and less in the later stages when the focus is on putting the plan into action. Similarly, the “less-capable” people work less in the planning stages and more in the later stages. This is an important benefit with plan-driven methods as project teams in the military often consist of people with various level of experience and various levels of skills. Personal experience from the Norwegian military is that rating skills of personnel is difficult. Some individuals are extremely talented in some disciplines, but show “lack of skills” in other areas. While some degree of unique, visionary people is great, the author expects difficulties in projects where they numbers exceed the critical mass, in the same way as he would expect problems to occur in a project consisting of a large amount of junior individuals. Boehm and

Turner (2003) explain that it is an unavoidable fact that 50 percent of the personnel is below average in any given discipline (or more mathematically correct; below median), and that access to really premium people is limited. The military organizations are designed to work with different compounds of personnel and should be realistic when selecting methodology. A “critical mass” is an inaccurate estimate, but since the military organization has a high rate of turnover, selecting a project methodology dependent of high amounts of unique people is likely to result in problems.

2.5.8 Summary of Methodology home grounds

The table below is meant to summarize the main points from the described methodology home grounds:

Project factor	**Plan-driven**	**Agile**
Requirements	Well understood in early development, pre-specifiable Best suited in projects where requirements are relatively stable.	Emergent, changes often. Well suited for frequent changes in requirements also in late development.
Knowledge	Dependent of documented knowledge.	Dependent of tacit knowledge and effective communication.
User-involvement	Mainly during the requirements phase and late testing.	Dedicated user representatives an essential part of the team of developers through a project.
Plan/planning	Plan in focus. After initial planning the focus shifts from planning to execution.	Planning in focus, open for change. The plan itself is useless after short time.
Size	Effective for large teams especially where developers are not collocated. Ineffective for small teams.	Works best in smaller teams, where developers and customers are collocated.
Criticality	Preferred for systems that needs high assurance, however also useful for less critical systems.	Best suited for less critical systems.
Culture & Personnel	Works best where people appreciate framework, policies and procedures or "thrive on order". Designed to work with less share of "premium people", but will work better with "premium people".	Works best where people appreciate high degrees of freedom or "thrive on chaos". Dependent of a critical mass of "premium people".

Table 2-2 Plan Driven Development vs Agile development

Adapted from Boehm and Turner (2003) and Cockburn (2000).

2.6 The military and software development

The US Department of Defense (DoD) is probably the world’s largest and most experienced procurement agency regarding software and have had a huge impact on how software has been developed. In 1980 the DoD issued the DOD-STD-2167, which demanded contractors to follow the linear waterfall model. NATO, governments and important private enterprises followed similar paths. The linear waterfall model had as mentioned a depressing efficiency rate, and in 1988 a new standard was issued, (DOD-STD-2167A) which was meant to encourage iterative and incremental development (Larman, 2004). However, this standard did not lead to any immediate improvement as it was interpreted as a shorter version of the 2167 standard (Larman, 2004). Several different standards have been issued, but no standard has been able to solve the problems in software development.

The National Defense Authorization Act Section 804 (NDAA 804) from 2010 has been referred to as a potential breakthrough for agile methods in the military. The NDAA 804 mandates improvement of DoD software acquisition processes and it encourages early and continual involvement of users, multiple executed increments or releases of capability and early, successive prototyping to support an evolutionary approach (Bellomo, 2011). While this may look like a demand for agile development, it can also be interpreted as a demand for incremental and iterative development which has been encouraged since the 2167-STD from 1988.

2.7 Agile Methods – a Silver Bullet or a Temporary Hype?

Although agile methods are still in their infancy and little academic research is available, prelimenary results are positive and the methods already enjoy great enthusiasm among developers. However, this enthusiasm is not shared by everyone. Rakitin (2001) is very unpleased with the agile hype. He claims that while software development is gradually becoming a a respected field of engineering, the agile hype hampers this progress. Ratikin believes that the agile advocates have a hacker-mentality, who when facing problems instead of working with them systematically do whatever necessary to immediately produce something that appears to be working. Buresh (2008) believes that although the goals of the agile manifesto are commendable, the whole field of software engineering is currently in a “state of flux” and therefore any significant results from the agile methods are for the present likely to be at best marginal. Boehm and Turner (2003) do not believe that the agile methods are silver bullets that can solve the problems of the software industry, but they refer to the methods as potential lead bullets, that can slay normal wolves, but are powerless when facing the “warewolf” projects. They argue that no methodology is fit for every situation, and that plan-driven and agile methodologies have different home grounds. The agile methods are excellent when it comes to delivering fast value to the customer, the plan driven methods are excellent when big workloads need to be carried out, when documentation is important and when lives are at stake. The authors argue that the key is to find the right balance, and to tailor a methodology to a project.

2.8 Chapter Summary and the Research Questions

The software industry seems less predictable than other industries, such as e.g. the hardware industry and therefore needs different management and methodologies. When this industry was still in its infancy, linear processes from other industries were adopted and the linear waterfall model has a dubious track record. Modern software development use plan-driven models and although they are often labeled as “waterfall” development, the literature suggests that modern projects are not managed strictly linearly, -feedback loops, incremental and iterative development is utilized also in plan-driven development. Plan-driven methods are described as heavyweight methodologies, meaning that the development process is supported by numerous control elements. This assures that heavy workloads can be divided on large amounts of workers which are not necessarily co-located and these heavyweight methods have advantages regarding proper documentation, which is important e.g. when software with severe criticality is developed. The drawback is that these methods can be very bureaucratic and ineffective (especially on small projects) and requirements are often locked in the early phases. The agile methods are lightweight methodologies and can be tempting alternatives especially in small projects. They rely less on plans, which means that they “welcome change” in e.g. requirements and technology even in late development, if this is needed to achieve good products. This gives a great potential for close collaboration with customers through a project. However, opening up for change comes at a cost. Agile methods can be problematic, cost estimation can become more complex, the development relies on tacit knowledge, the absence of a plan requires frequent meetings, the development is often poorly documented, projects can be vulnerable (especially to turnover of personnel) and size is believed to be problematic. It could seem as if the agile methods make life easier for programmers, but not necessarily for the project managers.

The literature review presents some interesting theories and the author is eager to see if there is a connection between the theories from the literature review and real life experiences. This will be attempted by asking the following research questions to the Norwegian Armed Forces' unit which handles software procurement and to one of their agile vendors.

1. What are the characteristics of the methodology applied by the Norwegian Armed Forces and of the methodologies applied by their agile vendors?
2. To what extent are projects pre-planned? Are plans followed mechanically or used as a basis for continuous modifications as projects progress?
3. What are the experienced challenges and critical success factors with the applied methodologies?

3 Research Methodology

Questions of research methods often result in a debate between qualitative or quantitative methods, but according to Guba and Lincoln (1994) the question of methods are of secondary importance to the question of which paradigm is applicable to research. Therefore the thesis will briefly introduce three paradigms in a simplified way before research approaches are presented. When these terms are properly introduced, the method of choice, the case study, will be investigated and explained.

3.1 Competing Paradigms and Underlying Research Philosophy

A paradigm explains the nature of the world for its holder and the possible relationships between the individual and the world, or more explicit: “the basic belief system or worldview that guides the investigator, not only in choices of method, but in ontologically and epistemologically fundamental ways” (Guba & Lincoln, 1994, p. 105). Ontology is the branch of philosophy that poses the question “what is the form and nature of reality and therefore, what is there that can be known about it?” (Guba & Lincoln, 1994, p. 108) while epistemology is the branch of philosophy that asks “What is the relationship between the knower and what is known? How do we know what we know? What counts as knowledge?” (Krauss, 2005). Krauss (2005) summarizes; “Epistemology, ontology and methodology are intimately related; ontology involves the philosophy of reality, epistemology addresses how we come to know that reality while methodology identifies the particular practices used to attain knowledge of it”. Guba and Lincoln (1994) name four major competing paradigms; positivism, post-positivism, critical theory and constructivism. Post-positivism addresses the identified problems of positivism, but the author will treat these two paradigms as one; positivism. This is undoubtedly a simplification, but several additional paradigms exist and this presentation does not intend to explore paradigms in detail.

3.1.1 Positivism:

<u>Ontology:</u> The positivist paradigm is often labeled as “naive realism” and assumes that there is one reality, “the way things really are” and that universe operates by laws of cause and effect that can be seen and understood if we apply the scientific method.

<u>Epistemology:</u> The paradigm regards the researcher and the research object as independent of each other and therefore the researcher can study an object without influencing it: “the researcher is independent of and neither affects nor is affected by the subject of the research”

Remenyi et al. 1998:33 forwarded by Saunders, Lewis, and Thornhill (2009, p. 114). Methodology: Hypotheses are stated and empirical tests are used to verify or refute these. Deductive reasoning is used to postulate theories that can be tested by further research (Burell and Morgan). Any influence from the researcher is a threat to the research validity and are avoided by prescribed procedures. Replicable findings are regarded as “true” (Guba & Lincoln, 1994). Quantitative research is often associated with the positivistic paradigms (Krauss, 2005). Mathematics, physics and chemistry are well suited for precise measurement (quantitative research) and are generally known as “hard” sciences while the results of social sciences are usually less quantifiable and therefore regarded as “soft sciences” (Guba & Lincoln, 1994) which leads us over to alternative paradigms. Generally, research is often viewed as either positivistic or phenomenological, and the two following paradigms belong in the latter category.

3.1.2 Critical theory:

Ontology: Guba and Lincoln (1994) use the term “historical realism” to label critical theory. At first no reality existed, but over time social, political, cultural, economic, ethnic and gender factors have crystallized into structures that are now regarded as real. While positivism is value-free, value is essential for critical theory. Critical theory observes the world through a political lens, and focuses specially on groups with power; e.g. politicians, men, and the capitalistic system in general. A critical theory differs from traditional theory in the way it serves a practical purpose. Critical theory seeks human emancipation, and the goal is to free humans from the historical structures that now enslave them.

Epistemology: The researcher and the research object are interactively linked, and therefore the values of the researcher are bound to influence the research, and the research is likewise bound to influence the researcher.

Methodology: critical theorists use analysis (historical, situational, textual) and qualitative interviewing.

3.1.3 Constructivism:

Ontology: Guba and Lincoln (1994) use the label “relativist” for the paradigm. The paradigm is often criticized for denying that a reality exists, that everything is relative, but according to

Kanselaar (2002) it only denies that we can rationally know a reality beyond our own experience. Our perceived reality is subjective and experiential.

The paradigm encourages researchers to construct their own knowledge instead of copying it from an authorities (such as textbooks, teachers etc.) and to learn in realistic situations rather than formal, decontextualized situations (Kanselaar, 2002).

Epistemology: The researcher and the research object are assumed to be interactively linked, and findings are created as research proceeds. Knowledge is relative (subjective) and therefore not absolute. While a positivist researcher observes the world through a one-way mirror, the constructivist recognizes himself as a part of it. The research approaches is holistic, and findings are not necessarily generalizable but can describe one specific organization, event, or even individual. While the positivist's goal is prediction, the constructivist's goal is to gain understanding.

Methodology:

Qualitative (primarily); open for interpretation; dialectical; contextual factors are described.

A summary of research paradigms follows below in Table 3-1 Summary of Paradigms, Ontology, Epistemology and Methodology.

Basic Beliefs	Positivism	Critical theory	Constructivism
Ontology (nature of reality)	One reality, knowable within a specified field of probability. -Fixed, stable, observable and measurable.	Rejects cultural relativism; recognizes that various versions of reality are based on social positioning; conscious recognition of consequences of privileging versions of reality.	Relativism, multiple socially constructed realities.
Epistemology (nature of knowledge; relation between knower and would-be known)	Objectivity is important, the researcher manipulates and observes in a dispassionate, objective manner.	Interactive link between researcher and participants; knowledge is socially and historically situated; need to address issues of power and trust. Value mediated findings.	Assumed interactive link between researcher and participants; values are made explicit; created findings.
Methodology (approach to systematic enquiry)	Quantitative (primarily), interventionist, decontextualized.	Qualitative (dialogic), but quantitative and mixed methods can be used; contextual and historical factors are described, especially as they relate to oppression.	Qualitative (primarily); open for interpretation; dialectical; contextual factors are described.

Table 3-1 Summary of Paradigms, Ontology, Epistemology and Methodology

Adapted from Guba and Lincoln (1994) and Mertens (1998)

3.1.4 Writing under a Paradigm

With these three major paradigms explained, the next step is to select a paradigm for the thesis. All three paradigms can undoubtedly be applied for a thesis on agile methodology. Positivistic research is likely to result in interesting results, and could have been applied in an attempt to investigate and compare very specific attributes with two methodologies, but it would be time-consuming and demand research of very high precision. Software development methodologies are very complex and to achieve the controlled environments necessary to give results without serious validity problems are beyond the scope of this thesis. Critical theory could have been applied, especially if the purpose was to describe social structures and the distribution of power in traditional and agile project environments. However, the author did neither wish to focus on power structures nor see it as his task to “free” workers from the structures that “enslaves” them and the political lens was soon abandoned. The constructivist paradigm seems more appropriate. Both traditional and agile methods are described, and one method is not likely to be neither a solution nor a failure in the field of software development, it is all relative to the alternative; traditional vs agile. The reality we perceive is subjective and experiential. Results of research might be difficult to generalize, because the same results might not be found when studying different research objects, even though they appear to be similar. Thus, the results are open for interpretation and not absolutes. As Krauss (2005) explains constructivism; “knowledge is established through the meanings attached to the phenomena studied”.

3.2 Research Strategies

The question of paradigm has been presented and the next step is to describe the different research strategies, methods and research design. There are various ways to connect these terms. Bryman and Bell (2011) explain the connection in the following way: Research strategy deals with the general orientation of research, (such as qualitative vs quantitative, deductive vs inductive etc.) while research method is the various techniques for data gathering (such as interviews, document reviews etc.) while the research design is the framework used for collection and analysis of data (such as the case study). A note can be taken of that Guba and Lincoln (1994) and Mertens (1998) understands the method term differently (see Table 3-1).

3.2.1 Qualitative or Quantitative Research

Approaches in the social sciences are often understood as either quantitative or qualitative. Quantitative research presume that things can be measured, counted and compared, while qualitative research presume that individual pieces of evidence are not directly comparable since the observations cannot be precisely measured (Gerring & Thomas, 2011). Quantitative researchers deal with numbers which can be directly compared while qualitative researchers use words and their results cannot necessarily be precisely measured or compared. There has and still is a debate between the followers of the two approaches, and many attempts have been made to explain the division between quantitative and qualitative approaches, however most work has been published by authors belonging in either the quantitative or the qualitative camp, and both camps have traditionally regarded the approaches as mutually exclusive and fundamentally different strategies. According to Bryman (1984) and Creswell (2013) a third camp exists, where scholars have debated how the two approaches can be combined, referred to as mixed methods, but the debate is still ongoing and it has not yet resulted in any commonly accepted guides for researchers.

	Quantitative	**Qualitative**
Relationship between researcher and subject	Distant	Close
Researcher’s stance in relation to subject	Outsider	Insider
Principal orientation to the role of theory in relation to research	Deductive, testing of theory	Inductive, generation of theory
Research strategy	Structured	Unstructured
Image of social reality	Static and external to act	Processual and socially constructed by actor
Nature of data	Hard, reliable	Rich, deep

Table 3-2 Differences between Quantitative and Qualitative Research

Adapted from: Bryman (2003, p. 94) and Bryman and Bell (2011, p. 27).

The table is a summary of the differences between qualitative and quantitative research, and while quantitative research appears to be positivistic of nature, qualitative research appears to be more subjective, which points in the direction of non-positivistic research (phenomenological). Gerring and Thomas (2011) explain the difference like this: “Quantitative tools help us compare and hence to generalize; qualitative tools encourage us to differentiate”. With the topic of the thesis in mind, qualitative methods might deliver data that are unfit for generalization, but the well suited to gain understanding of the differences between traditional and agile software development.

3.2.2 Linking Theory and Research -Deductive or Inductive Reasoning

Deductive reasoning begins with an assumption or a theory. The theory is narrowed down into more specific hypothesis that can be tested before the hypotheses are addressed by systematic observations, which test the hypotheses with specific data, and the initial theory is either confirmed or refuted based on the outcome of the tests. Inductive reasoning works the opposite way; starting with specific observations which then leads to a search for patterns which again leads to a tentative hypotheses and a potential broader generalization (or conclusions) if theories are developed. For this thesis the inductive approach has been selected, which is not a separate decision given that a constructivist stance and qualitative methods has been chosen and no propositions are present. However, the research questions are a result of the theoretical review, which can be argued to be a deductive process, but since no propositions are presented, the researcher sees the thesis as generally inductive. The difference is graphically explained in the figure below (deductive to the left and inductive to the right).

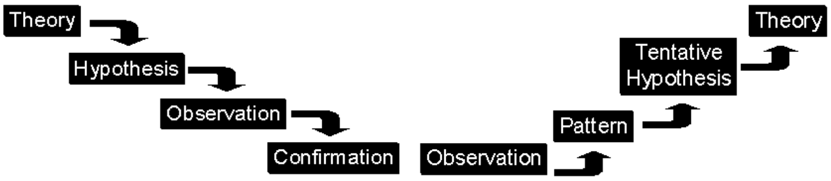

Figure 3-1 Deductive vs inductive reasoning

Source: Trochim (2006)

3.3 The Research Design – Case Study Research

Baxter and Jack (2008, p. 544) describe the qualitative case study as “an approach to research that facilitates exploration of a phenomenon within its context using a variety of data and sources”. They explain that this ensures that the issue is not explored through one lens, but rather a variety of lenses which “allows for multiple facets of the phenomenon to be revealed and understood”. There are two prominent authors who describe case study methodology thoroughly: Robert Stake and Robert Yin (Baxter & Jack, 2008; Tellis, 1997). While both authors are said to provide excellent guides to case study research, there are some differences and the author of this thesis will endeavor to apply the most updated framework described by Yin (2014). Yin’s definition of the case study is:

> A case study is an empirical inquiry that investigates a contemporary phenomenon (“the case”) in depth and within its real-life context, especially when the boundaries between phenomenon and context may not be clearly evident (Yin, 2014, p. 16).

This definition emphasizes that case studies investigate a present phenomenon and that instead of trying to achieve control over the environments, the phenomena are investigated in their natural context, especially when phenomena and context are hard to separate. Woodside and Wilson (2003, p. 493) criticize this definition and present their broader definition of case study research: an “inquiry focusing on describing, understanding, predicting and/or controlling the individual (e.g. process, animal, person, household, organization, group, industry, culture or nationality)". The author of this thesis does not see any fundamental differences between the two different definitions, and Yin (2014) definition seems appropriate; the phenomena will have to be investigated in their natural context and the software development methodologies and the investigated organizations are hard to separate as the borders are vague, but the process of data gathering will not be limited to contemporary events and in that aspect the thesis will lean against Woodside and Wilson's definition. Bryman and Bell (2011) point out that the case study is neither a research strategy nor a method, but a research design. The strategy is either qualitative or quantitative; the method is the technique for data gathering while the research design is the framework for collection and analysis of data. Case studies leans traditionally more towards qualitative research strategies but can also involve quantitative research (or both), and the design is open to a range of methods for data gathering.

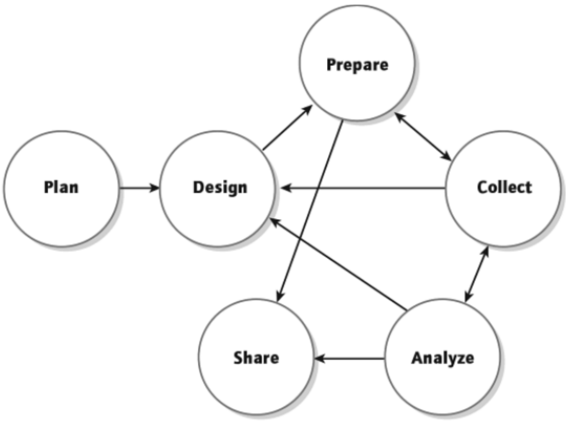

Figure 3-2 The Case study process

Source: (Yin, 2014, p. 1)

The figure above shows Yin's model of the Case study research process which will serve as a basis for the research framework of this thesis.

3.3.1 Plan and design

Case studies come in various shapes, and before the data gathering process started, a few important design attributes had to be chosen.

Exploratory:	This type of case study is used to explore those situations in which the intervention being evaluated has no clear, single set of outcomes.
Descriptive:	This type of case study is used to describe an intervention or phenomenon and the real-life context in which it occurred.
Explanatory:	A case study which seeks to explain the presumed causal links in real-life interventions that are too complex for the survey or experimental strategies.

Single case:	One case of analysis. Five rationales for single case design: critical case, extreme/unique case, representative/typical case, revelatory case or longitudinal case.
Multiple case:	A multiple case study enables the researcher to explore differences within and between cases. The goal is to replicate findings across cases. Because comparisons will be drawn, it is imperative that the cases are chosen carefully so that the researcher can predict similar results across cases, or predict contrasting results based on a theory.
Embedded:	A single-case might involve more than one unit of analysis. This occurs when, within a single case, attention is given to a subunit or subunits.
Holistic:	If the case study only examines the global nature of an organization.

Table 3-3: Case study designs

Adapted from Yin (2014)

An exploratory case study usually means to collect data first, and then try to investigate whether patterns can be found in the data material, and is typically used to answer questions like *what are the ways of developing software*? A descriptive case study requires deeper analysis and requires a theoretical review to direct the data collection. It is useful to answer questions such as *what are the results after the outsourcing process?* The explanatory case study involves deeper analysis and seeks to explain loops of causality such as: *why did the outsourcing process lead to decreased short term profits?*

This thesis presents research questions based upon the literature review, and the researcher has some expectations to what he will find and therefore it is not exploratory of nature. The researcher does not believe that he is able to analyze the data thoroughly enough to investigate direct loops of causality. Although the analysis might indicate cause and effect relationships, any data and analysis is unlikely to be robust enough to rule out alternative explanations. The question *How do the Norwegian Armed Forces and agile vendors approach methodologies for software development* is perceived as descriptive of nature hence the researcher regards this thesis as a descriptive case study.

The next step is to choose whether the case study is a study of a single case or multiple cases. Yin generally warns researchers to avoid single case studies unless one of the five rationales presented in Table 3-3 are fulfilled, because the single case study design is very vulnerable if a case turns out to be different than expected early in the research process. He compares single-case studies with research based upon one single experiment (Yin, 2014) which can in some cases be justified, but should generally be avoided. The evidence from multiple cases is often considered more compelling, and the overall study is considered to be more robust (Herriot and Firestone, 1983 forwarded by Yin (2014)). Yin's warning is noted and since the first research question will be better answered by examining contrasting methodologies, the multiple-case design is chosen. However, several cases usually indicate more use of resources (time and money) and in order to keep within the time frame a two-case design is chosen. Yin suggests that the cases, -the units of analysis should be chosen carefully, and either be chosen because similar or contrasting results are expected. The unit of analysis is usually individuals, small groups, organizations, partnerships, communities, relationships, decisions or projects (Yin, 2014, p. 35). For this thesis, two organizations which are assumed to have different approaches to software development are chosen. The first organization is the Norwegian Defense Logistics Organization's Information and Communications Technology (NDLO ICT) which is thought to represent traditional approaches to software development. The second case is Teleplan Globe, a software development company which in most of their projects acts as a vendor. Teleplan Globe is within the Defense known for agile methods, more precisely Scrum.

The last design attribute to be chosen is a *holistic* or an *embedded* design. The holistic design has been chosen (as it looks at two individual cases as a whole, rather than investigating different subunits). A multiple case design with embedded units would perhaps be more robust, but this decision has been taken on the grounds of the available resources regarding the thesis' time limit. Defining the case study design can be a challenge even for experienced researchers (Yin, 2014). This thesis threats the individual organizations as both units of analysis and cases. However, any major conclusions will be pitched at the level of the Armed Forces, rather than at the level of Teleplan Globe and it could therefore be argued that the Armed Forces acts as unit of analyses, while both cases act as units of observation, however, the researcher sees this as an issue mainly as an issue for semantics and that this interpretation is unlikely to affect the case study report with any significance.

3.3.2 Prepare

The preparation phase started when the researcher was gained approval to join the Norwegian delegation on Exercise Bold Quest where he acted as battle-captain in the Norwegian Head Quarters. The role can be described as a uniformed organizer who kept track of personnel, ongoing experiments and the person who facilitated the daily feedback sessions between field operators and team of agile developers. The exercise was held in various locations in North-Carolina in the period June 7th to June 22nd 2013. It was here the researcher was first introduced to agile development in practice. A log of events was written, and some brief comments added to a personal diary, but the writing of this thesis did not start up properly before late December and the thesis' focus pivoted repeatedly in the meantime. Although this exercise was initially meant as a session of participant observation, the researcher feels that it served as a source of preparation rather than as a source of data and evidence. It should be mentioned that it was during this exercise the researcher was first introduced to Teleplan Globe employees and the person who later provided access for the interview session at Teleplan Globe.

According to Yin (2014) the preparation phase includes gaining skills as a case-study investigator and conducting a pilot case. Instead of investigating an isolated pilot case, a round of five pilot interviews was carried out at NDLO-ICT near Oslo (Kolsås) on January 13th 2014. The interviews were more like inquiring conversations rather than structured interviews (see details in Table 3-4 Table of Informants). Several lessons were identified during five hours of unstructured interviewing. Some lessons were very concrete, such as the importance of recording the interviews which frees the researcher from writing down answers and allows him to focus on asking good questions. Another lesson was the importance of a thoroughly prepared interview guide. Unstructured in-depth interviews are a way of qualitative data gathering, but the author does not believe it is beneficial for a novice researcher. Although notes were written during the conversations, only parts of the conversations could be recalled when the notes were digitalized. Therefore, semi-structured interviews were chosen as main source of data gathering for the next rounds of interviews. It seemed beneficial to have a structured core, but also necessary for also the researcher to follow up unexpected answers to gain the necessary depth of understanding. The first interview session revealed that some of the interviewees had not thoroughly reflected on the questions before they were asked by the researcher. As a consequence the researcher saw it as beneficial that all interviewees were introduced to the thesis' main questions before the actual interviews were conducted. It was also noted that the informants had limited amounts of spare time during the work day, and this further supported the idea of introducing the respondents to some of the key topics before the interviews. The researcher had an expectation of that this would allow interviews to be conducted more efficiently as the respondents would have to spend less time in order to consider unexpected questions. The pilot test also revealed that the possible respondents had limited amounts of time available, which suggested that any written material sent out to the informants needed to be brief and precise and that the individual interviews had to be conducted efficiently. The pilot test also led to further refinement of the theoretical review and the research questions.

3.3.3 Data collection (research methods)

For this thesis individual interviews serve as the main source of evidence. In order to minimize language barriers all interviews were conducted in Norwegian since this was the mother tongue of both interviewees and the researcher. The informants had been sent a participation information sheet (Appendix B) and the interview guide (Appendix C) prior to the interviews. The participant information sheet described the thesis questions and provided some practical information while the interview guide showed the basis for the interviews. These were sent by email to the respective facilitators in the two organizations five days prior to the scheduled interviews, and were distributed to potential candidates (see Appendix B – Participant Information Sheet and Appendix C – The Interview Guide). The very last interview was not originally planned, but a result of the snowball sampling strategy (explained in section 3.3.4). The interviewee was presented these documents in printed form only two hours before the interview started, and she needed to attend a meeting in between, but she stated that this was not a problem as she was familiar with the topics for discussion. Both documents can be argued to have affected or even polluted the gathered data, but the researcher endeavored to write them in neutral language in order to minimize the effect of reflexivity (informants give the answers they think the researcher wants to hear) and biased language. Questions were asked open-ended, and informants were allowed time to think whenever they felt it was necessary. The researcher rephrased questions in situations where it seemed beneficial, and the aided memory of the interviewees were sought whenever the informants were asked questions they were unable to answer after a few moments of thought gathering.

The pilot round of interviews had revealed that it would be beneficial to use a voice recorder (presented in 3.3.2), and this was used for the second round of interviews at NDLO ICT. However, some notes were also taken during the interview, partially as backup in case of technical problems, but also in order to emphasize and draw models the informants shared on blackboards. Prior to the third round of interviews at Teleplan Globe the facilitator was reluctant to audio recording and offered a little more time for each interview in return. This was of course respected. However, it demanded a higher degree of structure during the interviews and answers had to be written down in a way so that they would still make sense when the researcher had time to examine the data material.

Interviews are great data sources as they focus on the case directly and provide deep insight, but they also have some drawbacks researchers need to be aware of such as biased answers, poorly articulated questions, inaccuracies due to poor recall and reflexivity (Yin, 2014, p. 106). This could be minimized by trying to conduct interviews in the spirit of the positivistic researcher, to observe through a “one-way mirror”, but since the number of informants, the time and budget was limited, and the researcher modestly has to admit that he is not an experienced interviewer, the interviews were held in the spirit of the constructivist, where the researcher accepts that he or she becomes a part of the research process.

The research questions were used as a foundation for the interview guide, and the researcher chose to also include a few questions which did not necessarily answer the research questions directly, but which were included in order to gain a broad understanding. The interview guide was used as a starting point for the interviews, and the interviews were conducted as a conversations rather than strict questionnaires. The interviews were planned to last around 40 minutes, but they varied from 35 minutes to 90 minutes, depending on the interviewees’ other arrangements. Some interviews, typically the first interviews during the day were concentrated around the first questions from the interview guide, and due to time restrictions, the last questions were allocated much less time. When this happened, the consecutive interviews were adjusted to focus especially on the questions which had been allowed little time. All choices regarding interview technique can affect the data material, and although the researcher actively tried to express questions and documents as neutrally as possible, the effect of biases and reflexivity cannot be ruled out.

A second source of evidence is documentation, more specifically two reports written by the Norwegian Research Establishment (FFI). The first report is a descriptive case study written on the cooperation between the Army Special Forces and Teleplan Globe. The case study does not focus on software methodology, but rather on the culture within a uniformed unit which worked together with a flexible vendor. Although the report does not focus on agile methodology, parts of the development process are well documented and it establishes a timeline. The successful customer interaction between agile software engineers and uniformed customers is outlined in the report, and although the study mainly investigates and accredits the uniformed customers, the described process stands out like a textbook example of agile development. The other report is the result of a research work which was requested by the NDLO. It maps bottlenecks and the level of competence regarding NDLO's investment processes.

Reports present secondary data, which can represent biased selectivity of material and reflect biases held by the authors (Yin, 2014). The researcher tried to treat the material with care in order to minimize the effect of biases, although the material is to some degree bound to influence the researcher, which can be accepted when writing under the constructivist paradigm. On the other hand, reports are unobtrusive, -they are not results of this case study, and it describes a series of events in detail.

3.3.4 Data sampling

There are generally two umbrellas which capture two different approaches for data sampling; probability sampling and non-probability sampling.

Probability sampling

A probability sample is selected by using random selection so that each unit in the population has the same chance of being selected and in this way a representative sample is sought. The goal of probability sampling is to minimize the effect of sampling errors, which occurs when a difference between the population and the sample occurs (Bryman & Bell, 2011). Probability sampling is the domain of quantitative studies and positivistic researchers and a science of its own and will therefore not be discussed any further.

Non-probability sampling

Non-probability sampling is more common in qualitative studies and the term captures all forms of sampling that are not results of random selection in order to achieve representative samples of a population. Bryman and Bell (2011) describe three types of non-probability sampling; *convenience sample*, *the snowball sample* and the *quota sample*. A convenience sample is a sample that is chosen due to the researcher’s accessibility. The generalizability is problematic when using convenience samples since the researcher does not know whether the sample is representative for the population. Bryman and Bell (2011) emphasize that this does not mean that convenience samples should be avoided; the results might be interesting and although they do not allow definitive findings to be generated, they might serve a preparatory purpose for further research. Snowball sampling is actually another form of convenience sampling, but Bryman and Bell (2011) emphasize the difference because of it has attracted massive attention in research. The concept is that the researcher starts with contacting a small group of people that are relevant for the topic of study and uses this group to get in contact with other relevant individuals. Snowball sampling is not random selection, and it suffers from the same problems regarding generalizability. A third way of non-probability sampling is quota sampling, which is frequently used in marketing research. The idea is to produce a sample that reflects the population in such a way that relative proportions between categories (or quotas) such as gender, age, ethnicity, religious beliefs, etc. are maintained. Even though the sample is not a result of random choice, the results of quota sampling has a higher degree of generalizability than the other ways of non-probability sampling and Bryman and Bell (2011) refers to researchers that claim that quota sampling is almost as good as probability sampling. The sampling strategy for this thesis is a mainly snowball sampling, but it also has clear indications of convenience sapling. A few individuals that were assumed to be relevant were initially contacted and these informants were able to put the researcher in contact with individuals they thought would be valuable for the researcher. The researcher tried to some degree to balance the sample in a way so that stakeholders of different categories were represented; project managers, project owners, programmers and mercantile representatives, but it was only a partial success. This sampling process is likely to have affected the data material and ultimately the results of research which might indicate that the data material may be unfit for generalization. Another issue is that this is an educational qualitative study, hence time, capacity and funds are of limited availability. Therefore the relatively small sample of thirteen informants, i.e. the selection of informants is likely to affect the result of research and the results would probably have been different if another sampling strategy had been chosen.

#	Position:	Years of professional experience with ICT development:	Date of interview:
1A	NDLO ICT – Project Department – Mercantile expert	5	13.01.2014
1B	NDLO ICT – Project Department – Project Manager	20+	13.01.2014
1C	NDLO ICT – Project Department – Project Manager	20	13.01.2014
1D	NDLO ICT – Project Department – Project Manager	7+	13.01.2014
1E	NDLO ICT – Project Department - Mercantile expert	2	13.01.2014
2A	NDLO ICT – Project Department – Project Manager	20	12.05.2014
2B	NDLO ICT – Project Department – Project Manager	8+	12.05.2014
2C	NDLO ICT – Project Department – Project Manager	5+	12.05.2014
2D	NDLO ICT – Operative system Department –Engineer	8	12.05.2014
3A	Teleplan Globe – Managing Director	18+	13.05.2014
3B	Teleplan Globe – Managing Director	18+	13.05.2014
3C	Teleplan Globe – Project Manager	5+	13.05.2014
3D	Teleplan Globe – Project Manager & Product owner	13	13.05.2014
3E	Teleplan Globe – Project Manager & Product owner	11	13.05.2014

Table 3-4 Table of Informants

Job positions and titles were actually much more complex than the table indicates. They have been attempted standardized in order to make it more evident what their tasks was, and to give the informants some degree of anonymity. Therefore the specific job titles and senior titles have been left out from the table. Working experience was mentioned orally during interviews and relevant experiences have been added up by the author and some interviewees will probably feel that this is inaccurate. A “+” added to the working experience indicates considerable amounts of experience that is seen as valuable, but which is not considered directly relevant to their current positions. The sample consists of thirteen informants who participated in fourteen interviews (interview 1D and 2B is the same person). The sample has a limited size due to capacity. Three more interviews were originally planned, but were canceled due to colliding activities or emerging situations which needed immediate attention from the informants. Both units of observation are located in Bærum (close to Oslo) and required traveling and accommodation and therefore the interviews were conducted efficiently during three working days. Software is only one of several categories under the ICT umbrella, and although most informants worked with software projects, few projects were exclusively software projects. They were actually combinations of software, hardware and infrastructure. One project manager had infrastructure projects as his specialty and therefore had a portfolio which almost solely consisted of infrastructure projects. At the same time, he was highly familiar with the software development process, he had great knowledge of the organization and had spent time reflecting on topics presented in this thesis. His answers were of great value for this thesis and are included in the report.

The sample consists almost exclusively of project managers and this can have affected the data material. Ideally, programmers, developers, architects and software engineers should also have been interviewed, but the project managers had years of experience from positions in the lower levels of the hierarchy which is thought to partially compensate for the relatively uniform sample.

3.3.5 Analyze

All Data Analysis seeks to bring order to the chaos that naturally resides in the gathered material. For qualitative studies this means to conduct a systematic process in order to “boil down” the large amounts of data until the data can be used to provide insights to the research questions. Yin (2014) claims that the analysis phase of case studies is one of the least developed aspects of doing case studies, but he advices case study researchers to start with the questions and then read through the material in order to find evidence that addresses these questions.

Data from Interviews:

The interviews were conducted in three rounds and the gathered data material was different from each session. The notes taken during the first five unstructured interviews were partially digitalized, but key words that did not make any sense after the interviews were not transferred from the notebook. These interviews served mainly as preparations in order to gain a broader understanding of the topic, to adjust the literature review and to practice interviewing and little data from the first round is included in the report. The second round of interviews gave much better data material. The resulting audio files were transcribed word-by-word and stored on an off line hard drive. These transcripts were then shortened by the researcher and compared with the notes. Any superfluous words and sentences that were found irrelevant were then erased and these new files were stored on the mentioned hard drive. These text files were then printed on paper and put in separate folders. Each research question was allocated a distinct color, and the researcher read through the material and colored quotes and sentences that were found relevant to answer each research question with the respective colors. Although it was a time consuming process, it was found much more effective than early attempts done with spreadsheets in Excel. Round three of interviewing was done without audio recording. The resulting notes were digitalized immediately after the five interviews were completed, in an attempt to contain the meaning of those notes before meanings were lost because of a volatile memory. All resulting text files were printed and put in individual folders, and the described process with highlighters was repeated. When this process was completed, the framework for the individual case reports were established, and data connected to the research questions (topics) was transferred to respective chapters, before a vast job of structuring this material into something readable was done.

Data from Reports:

The researcher gained access to several reports during the writing of this thesis. Two of them are referred to in the report. Neither of the reports discusses approaches to software development in particular or presents a set of data that could be used in this case study report. Therefore they have not been used methodically. The FACNAV report was used mainly to understand the array of stakeholders, how they worked together and what the researchers believed were the most important factors which contributed to the success. The report concerning analyses of bottlenecks in the investment process at NDLO is used to support the empirical data presented, not as individual sources of evidence.

3.4 Quality of research

As previously indicated a case study has certain strengths; the design is useful when trying to understand and explain complex phenomena especially when the boundaries between the phenomena and context are unclear, as it might be hard to draw a line between development methodology and the units of observation. The design also has a few weaknesses, especially case studies like this one which relies on qualitative data. Yin presents four criteria which can be used to estimate the quality of research designs; *construct validity, internal validity, external validity* and *reliability*.

3.4.1 Construct validity

Construct validity requires that the researcher must identify operational measures for the concepts being studied, and Yin (2014) explains that this is especially challenging in case study research; investigators fail to develop a sufficiently operational set of measures and that subjective measures are used to collect data. This seems unavoidable for a case study which merely relies on qualitative data. Although the construct validity will never be excellent for case study research, Yin (2014) suggests that this can be counteracted by using multiple sources of evidence, have a draft of the case study reviewed by key informants and to establish a chain of evidence, which means that the reader of the case study should be able to trace the steps of research the whole way from data gathering through the process until ultimate conclusions are presented. The researcher tried to keep this in mind when writing the thesis, reports were actively searched for to be used as additional sources of data and two of these were found relevant, but they are additively not used in a way which significantly improves the construct validity. Yin (2014) suggests to have key informants review the report in order to improve the construct validity, and in Yin's spirit the informants were invited to review the thesis. However, feedback was absent and in retrospect it would have been more favorable to present the informants with a few slides with key points rather than a massive document. In total, this indicates that the construct validity of the thesis might be questionable.

3.4.2 Internal validity

The question of internal validity is applicable to explanatory studies such as this study where casual relationships are investigated; how and why did one specific event lead to another specific event. The investigator must gather sufficient proof to that "event a" led to "event b" and make sure that other important events are not overlooked (Yin, 2014). The researcher sees this as a common flaw in research projects; that causality and correlation are confused. Several tactics can be used to increase internal validity and one is to address alternative explanations (Yin, 2014). The researcher tried to bear this in mind, but sees this as more important in explanatory case studies which actively seeks causality loops. This case study is descriptive and tries to reproduce connections as they are experienced by the informants. Connections might be indicated, but the limited data material and the analyses process is not fit to claim direct cause and effect relationships.

3.4.3 External validity

External validity deals with the problem of settling to which degree a study's findings are generalizable beyond the unit of analysis. This case study investigates two units that might be either unique or representative compared to similar units, but because only two cases investigated, the results are probably unfit for generalization beyond the two cases that are investigated. The analysis can however provide useful insight for other units who face similar challenges as the units of observation, or future researchers investigating the same phenomena.

3.4.4 Reliability

Reliability deals with the question of whether another researcher who followed the same procedures as described by the investigator would draw the same set of final conclusions (Yin, 2014). The intention of reliability is to secure that the effect of errors in research and cognitive biases held by the researcher are minimized. Although the researcher has aspired to describe the process accurately and tried to avoid biases and errors, the researcher must modesty acknowledge that he lacks research experience and there is a possibility that other researchers would come to different conclusions, however, the researcher has been aware of this and tried to mitigate this by constantly challenging personal biases, discussing the thesis with experienced researchers, and discussing with acquaintances with different views on the topic. This has been helpful and has undoubtedly contributed to a less dogmatic view on key issues presented in the thesis.

3.5 Ethical considerations

The ethical principles presented by Bryman and Bell (2011) were used as a guide for the study; no harm to participants, informed consent, no invasion of privacy and no deception of participants.

No harm to participants is a principle which seems to be self-evident, as harming participants is usually regarded as unacceptable. However, Bryman and Bell (2011) raises the question; what is harm? It can be obvious such as physical pain, but harm can also be psychological which might be less concrete; stress, harm of self-esteem or subjects can be encouraged to perform acts that they are not comfortable with. The author believes that a researcher which interacts with people can never guarantee no-harm to participants, but it should be kept at a minimal level and this served as a principal guideline for this research.

Informed consent demands that the respondent gets enough information to take an informed decision whether they would like to participate, which is a principle that is jeopardized e.g. when covert participant observation is used to gather data. The researcher behind this thesis stated his intentions upright in a participation information sheet and also shared the interview guide prior to the interviews.

Invasion of privacy is an area of ethical concern which relates to in which degree invasion of privacy is acceptable. Bryman and Bell (2011) explains that interviewees who have agreed to participate in a study frequently will refuse to answer certain questions, and either because the questions invade private realms or deals with matters that the interviewees wants to keep out of public interest. The author did not see this as a likely problem; the matters discussed barely involve any elements from the private sphere, but the researcher was prepared to stop recording or skip certain questions in case such unlikely situations occurred.

Deception in research is when researchers represent their research as something other than it really is, and although deceptive methods historically have led to several famous and shattering reports (such as Milgram’s 1963 obedience experiments), the ethics of such research is questionable. The researcher must carefully weigh the gains versus the harm done to participants. Human suffering must be minimized. The researcher has applied forthright methods for data gathering and “trick-questions” which seeks sensational answers for the thesis were avoided.

3.6 Personal Information Protection and Data Management Routines

In this thesis all actors were anonymized and all subjects have during the data gathering process been referred to with an alphanumeric code, rather than their names. The researcher noted the link between these codes and names of the respondents in a personal notebook, as this allowed informants to be contacted for necessary questions. This could potentially jeopardize the anonymity of the respondents, but it has been in the researcher's control at all times.

The sample sizes are limited and one of the organizations has a very limited number of employees, therefore the informants are likely to recognize their own statements, and probably some statements from their colleagues. The specific job titles have been left out, but with a very limited sample of thirteen individuals, it seemed unavoidable that colleagues could identify each other's statements. It is not viewed as unproblematic, but the researcher argues that is on an acceptable level, where the gains of research surpass this potential inconvenience. Data management routines and the purpose of the interviews were explicitly explained in the participant information sheet. The researcher has not gained specific approval from the Norwegian Social Science Data Services (NSD), as the project avoids storage of data which triggers an obliged license. All data from interviews (audio files, transcripts, notes, lists of names) have been stored either on a portable hard drive or in the researcher's private office in order to prevent any unintended sharing of material. All this material will be shredded or deleted when the thesis is evaluated, a grade is awarded and the final appeal deadline is exceeded.

4 Empirical results

Data from the two cases is presented in this chapter. It is presented following a framework which derives from the research questions. The subchapters will therefore be presented using this frame: Introduction of the case, Methodology, Plan or Improvisation and finally Challenges and Critical Success Factors with the applied Methodology. The headings are admittedly broader than the research questions imply, but the researcher has tried to build a robust foundation before research questions are answered.

4.1 Norwegian Defense Logistic Organization - ICT

The Norwegian Defense Logistic Organization (NDLO) is one of 21 operational units within the Norwegian Armed Forces. NDLO is responsible for the procurement and management/ownership of the material in the Norwegian Armed Forces and it is responsible for approximately 50% of the Norwegian defense organization’s budget. It is a diverse unit that has the responsibility for everything from procuring socks to maintenance of sophisticated weapon systems. They deliver services both to the Army, Navy, Air Forces and the National Guard. The NDLO consist of seven units; Land capabilities, Naval capabilities, Air capabilities, ICT capabilities, Joint service capabilities and Maintenance and Supply. The ICT capability unit has the responsibility for securing all the ICT systems that are used in the Army, Navy, Air force and National guard, as well as operational units. They are an owner and a manager of the IT systems. NDLO ICT has approximately 330 employees and consists of seven units: Staff, the Project Department, the Contract Department, the ICT Architecture and Security Department, the Operative System department, the Core Service Department, Secure Platform Department, the Network Department and Base Support Kolsås. When a new project is established, the Project Department in the ICT unit will be responsible for the execution of the project. They receive necessary support from the Contract Department who will write the strategy, the enquiry and the evaluation process together with the ICT Architecture and Operative System Department. It is a close collaboration between the units within ICT Capability. Numerous civilians, such as lawyers, engineers, economics and programmers, are employed by the NDLO and compared to rest of the Armed Forces, the NDLO capability unit is considered to be a segment with a high degree of business competence.

Source: www.forsvaret.no (accessed 13.05.2014)

4.1.1 Methodology

When the project managers from the Project Department were asked to describe the dominant methodology of NDLO ICT, the waterfall model and the “Prinsix” framework was referred to frequently. The boundaries between frameworks and methodologies are vague, and although Prinsix can be understood as a methodology, the majority of the interviewees saw it more as a general framework than a specific methodology. One project manager stated that Prinsix was too general to say anything concrete about the development process; it neither encourages nor excludes any concrete development methodologies. In other words, development processes could be linear or non-linear, it could practice integration of functionalities in late stages of development or it could practice incremental growth. It could facilitate iterative development or demand iron bound plans in early development, as the framework does not direct actual development methodology. A team of developers could use single-pass waterfall, iterative waterfall, Scrum, it was totally dependent of whatever method which was practiced by the contractor which was awarded a contract. However, the Armed Forces have an evident legacy, and when the same project manager was asked about the applied methodology, he answered that NDLO used waterfall, but it was “the version of waterfall where arrows points in both directions” (shown in Figure 2-7 “Real-life Waterfall”). This “iterative” waterfall model could in theory be flexible, but one project manager stated that due to the prevailing culture and a strong cultural inertia, the method was experienced as “rigid from start to finish”. Although it was described as rigid, it was not uncommon that project teams gained new knowledge during projects, and this knowledge often led to some alteration of work done in previous stages. Early work could be modified in later stages, as Prinsix was open to rework when this was required. However, it was said to be certain restrictions due to the legal business regulations, such as national regulations of tenders and contract awards. One example which was mentioned was if the requirements vendors originally based their tenders on had to be altered severely after a contract had been awarded, legal regulations could cause an investigation and a delay, or even a termination of the project. The same project manager said that he saw this as reasonable, because if several of important requirements had to be altered, this could indicate that the project would not fulfill its original purpose and “pulling the plug” could in some cases be the only right choice.

This thesis' questions concern software development methodology and an interesting discovery was that the project managers in the Project Department in NDLO ICT were located far away from where the actual development process took place. Several of them gave the impression that they acted as links between the customer (the Department of Defense) and the development organizations, and that their duty was to manage projects only on a higher level. Several of them explained that their roles were more as facilitators for other actors and that they were mostly involved in the phases before the actual product development started and when a project approached completion. They were not directly involved in any actual development unless obstacles occurred, or important decisions had to be made when a project reached important crossroads. The project managers also reported that their busiest periods during projects were in the early phases, actually in the stages before any actual development took place. When the product development started, they reported to be much less busy, and in theory they had a more relaxed atmosphere, but since they all had portfolios of multiple projects (from two to five) they usually had to attend other projects (which had suffered from a lower priority) immediately. The actual product development is principally outsourced to external vendors, who are either awarded a contract for a specific project or which has an established outline agreement with the NDLO ICT.

The Prinsix Framework

Prinsix standardizes the investment process and coordinates the many contributions from different actors in the investment process and ensures that the organization operates in accordance with policies, laws and regulations, and makes it possible to track progress. This is rational for large governmental organizations with varied portfolios of tasks and projects. Although it may be argued that Prinsix is a framework and not a methodology, a vast portion of the information gained from the interviews was based upon the interviewees' experience with the Prinsix framework, and therefore, a general overview of the framework is presented below. Prinsix is comprehensive and complicated, and to fully understand the framework takes years of experience and education. The following presentation is simplified and in some cases inaccurate, but included here to provide a basic understanding of the organization and the environment.

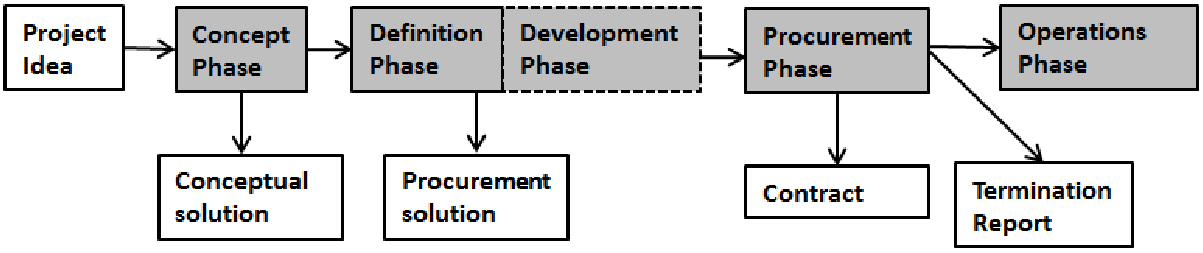

Figure 4-1 The Prinsix Investment Process

Adapted and translated from: http://prinsix.forsvaret.no (accessed 13.05.2014)

Prinsix is the framework for project management applied by the Norwegian Armed Forces. While white boxes indicate documents the gray boxes indicate phases. The starting point is the Project Idea, which initiates the Concept Phase, where potential solutions are evaluated. The Department of Defense issues an assignment to the Staff of Defense and manages the work with issuing the Conceptual Solution. Then the analysis of the requirements starts which tries to endure that solutions are not selected without a thorough process. This is followed by an analysis of alternative solutions. This leads to the writing of the Conceptual Solution document. The following Definition Phase results in the document Procurement Solution which is the decision basis for the Department of Defense when they decide whether a project is to be initiated or abandoned. If the requirements cannot be covered by standard available commercial material a Development Phase is required. During the Development Phase, tests and evaluation of different alternatives are thoroughly investigated, prototypes are developed and this leads to an update of the Conceptual Solution Document. The Development Phase can result in termination of a project, if the conclusion is that the project is not feasible. If it is not terminated, the Procurement Phase is initiated, tenders are issued, before contracts are awarded to vendors. When the software or material is procured, it is delivered to users and a Termination Report is written. A note should be taken of what Prinsix coins as the Development Phase is actually not the phase where products such as software is developed, this is done after a contract is awarded and Prinsix refers to this the Procurement Phase.

Source: interviews and http://prinsix.forsvaret.no (accessed 15.05.2014)

As a result of the snowball sampling strategy (see 3.3.4 Data sampling), the last interview was held with a senior engineer who worked as a team leader in the Core Enterprise Services of NDLO ICT. Her answers were interesting and in contrast to the answers from the project managers. When she was asked to explain which methodology she used, her answer was immediate; “all my work is Scrum-based”. She reported that she was all in all satisfied with the Scrum process, it worked great for her purpose, but it was in some cases actually experienced as rigid, and therefore she and her team also used Kanban for certain specific purposes. Kanban (a scheduling system for lean and just-in-time production) was a method which she found easier to tailor to her team’s needs. This team consisted of architects, programmers and her. She had neither learned Scrum nor Kanban in-house, it was something she had been introduced to and learned by a previous employer (an interactive communications agency). The team leader thought she was the only one to use Kanban at NDLO ICT. When asked about Prinsix, she said that Prinsix was of great importance for the Project Department, but personally she did not use it much. She and her team worked much more “on the ground” and interacted with customers and user representatives regularly.

NDLO ICT methodology and the iron triangle

When the first interviews were conducted at NDLO ICT in January 2014, the iron triangle was mentioned repeatedly. This was perceived as particularly interesting and therefore the later interviews approached the iron triangle more methodically. The answers were not uniform. One project manager expressed that product quality was the “sinister” variable since project performance was measured exclusively on time and cost. However, he added that if the security issues are considered as a part of product quality, this component of product quality was fixed but apart from security issues, product quality was difficult to measure objectively and probably therefore not used to measure progress. Two project managers gave different answers to the same question. They both understood perceived quality and cost as fixed, which indicated that time was the balancing factor. The rationale was that the product quality to some degree is defined when the requirements are written, and although much can happen technology-wise during the development, the quality is defined. One project manager gave a quick introduction to how the project’s cost limits were calculated. Cost estimates were calculated so that there should be an equal chance of exceeding and going below budget. The projects were always allocated a funding reserve which adjusted the numbers to 85% chance of delivery within budget, and 15% chance of exceeding. If projects exceeded this 85% frame it was necessary to apply for increased funding. This was not perceived as troublesome process – unless projects were of political interest, something which often happened when projects received media attention or attention in the Norwegian Parliament.

4.1.2 Plan or Improvisation

The Armed Forces have a standardized process for project planning and execution. Plans are initially written for the whole project, but they are not very specific regarding details in the initial phases. When the interviewees were asked about whether there usually were fundamental differences between early plans and the results of completed projects the answer was that in most cases there was a clear coherence; requirements are defined in the early phases and to a fair extent, the requirements result in the plan and the plan is the driver behind projects. It was, however, stated that specific technology was rarely defined in the requirements and neither should it be, as defining technology in the early phases was like asking for trouble. One informant said that planning with technology was "like shooting at a moving target" meaning that if you aim at the present technology you are bound to miss. Projects were reported to last from one to five years, (with an estimated average of three to four years), so defining technology early would mean that users would unwrap antique products, "straight from the box" when they were delivered. One interviewee says that even though you do not specify technology, vendors offer their services based on contracts, and although the technological requirements are not specified in the early phases, at some points irreversible decisions have to be made. Vendors need to know if they should use technology A or technology B and this might be difficult to reverse retrospectively.

The project managers experience that the Armed Forces is a large organization with a lot of parallel activities and that plans are altered often. There is a constant pressure on various resources and this frequently causes delays. One of the informants estimates that he usually has a fair understanding of what happens for the following three months, but everything beyond that is very uncertain. Another project manager has a portfolio of projects with the Norwegian defense industry, more specifically solutions for communication security. The project manager expressed that these projects are atypical. They are to a high extent regulated by contracts, and massive pre-work has been done in advance to reduce risks and uncertainties before contracts are written. He expressed that the defense industry projects are a bit special as they are more bound by plans than other projects, and that usually only minor, cosmetic changes are done to plans during projects.

One informant explained how NDLO ICT separates between “black” and “green” material. Black material is ordinary ICT material which is also commercially available for private enterprises, while green material is specific military equipment. He thought that NDLO ICT dealt well with “black” material, but that they repeatedly delivered outdated material in the “green” category. One of the reasons is that green material has unique demands regarding security and demands against environmental stress, often mentioned as “military specifications” or “mil-spec”. Another issue is that while black software material usually can be updated or corrected with patches or updated versions, this was difficult with the “green” material. The security organizations give their approvals to products in “in their complete form”, meaning that “green” material would lose its approval if it was modified or updated and to get new approvals could require years of casework.

4.1.3 Challenges and Critical Success Factors with the applied Methodology

One of the informants expressed that writing proper requirements was probably the number one challenge for the NDLO ICT and that although it is not declared a specialized discipline he thought it could be wise to do so. He believed that there was an importunate need for education of employees regarding defining of requirements, because smartly written requirements were a prerequisite for successful projects. This was supported by another project manager who stressed that requirements had to be written smartly. Rigid requirements would hold projects in strait jackets, while “bendable” requirements could allow crucial flexibility in the process.

Although the NDLO has a strict system for documentation and reporting, this does not necessarily mean that knowledge or experience is effectively shared between individuals or groups. During projects, vast amounts of documents are produced, but when a project is terminated (completed), most of the documentation is archived and said to be difficult to access. However, Prinsix demands a termination report to be written. This report is meant to gather lessons learned from the whole project team and would ideally lead to shared experience. It describes the execution process in detail, experience with time and cost and the experience and impressions with the involved vendors. The idea behind termination reports is mentioned as good, but in practice, they are written to reach the final milestones and one project manager expressed that they are rarely read after a project’s completion. A project manager expressed that the termination reports are difficult to access and unless you are interested in reading a termination report from a specific project, the reports are bothersome to access. One of the project managers expressed that there were in reality no established systems for knowledge sharing, and that knowledge management was one of their key challenges. He suggested that a SharePoint solution or with a searching function could be of great value. The current system where the reports were hidden away in various network folders were not regarded as satisfactory.

A report written by FFI (the Norwegian Defense Research Establishment) which investigates “bottlenecks” found within the NDLO supports this. The report concludes that cross-project learning does not happen and quoted interviewees claim that in reality documents such as termination reports are probably never read, but written merely to reach milestones. One of the report’s four recommendations is that “the NDLO needs to become a learning organization” (Gulichsen, Strans, & Steder, 2012).

One of the informants expressed that the cross-project knowledge sharing which actually took place relied on people’s personal networks and who they socially interacted with. It was also mentioned that there were some informally appointed “gurus” in the hallway which were frequently asked for advice when project teams faced problems.

The access to key personnel was mentioned as a limiting factor. In many cases certain field of expertise was “one man deep”, and if this person was unavailable, it could especially in the initial phases lead to delays. One of the project managers reported that he did not see this as a serious problem, but he admitted that he was probably affected by his five years of experience in the Project Department and had accepted “that was just how things were”. He experienced it more as a challenge than a problem. It was experienced as difficult to replace key individuals, and this was usually only done if people left the organization or were absent for more than six months, most typically when people left for parental leave.

A project manager explained that that when people were dedicated to a project they experienced rapid progress, but in the real world, the personnel (project managers, engineers, architects, mercantile experts etc.) all had their portfolios of projects. One of the informants had once experienced that his project was declared a high-priority project by the political level. He stated that he had been impressed when he saw what could be achieved when projects had access to dedicated personnel. However, he also stated that the NDLO ICT had 42 projects at the time, and while the two prioritized projects progressed rapidly, the remaining 40 projects suffered from delays. Another project manager expressed that the lack of dedicated personnel was experienced as problematic. Projects were usually dependent of work packages and individual contributions needed to follow a specific order. He continuously experienced that his projects frequently stopped completely until the right individuals could allocate time to work on his projects.

4.2 Teleplan Globe Ltd.

Teleplan Globe is a software company and a part of Teleplan Holding. Its core activity is development of software and systems that support operational functions desired by their customers. The Norwegian Armed Forces are undoubtedly their most important customer, but they also develop products for the Norwegian Police, the Norwegian National Security Authority, small and large private enterprises and to a limited extent some foreign military organizations. Their specialty is products for radio/frequency planning, network planning and geographical information systems. It is claimed on their website that their strength lies within high technical competence and their tradition for close collaboration with users throughout development. The company employs around 70 people and the majority of the employees, around 55 of them are software engineers. Their products are offered on the global market, but with a few exceptions, Teleplan Globe earns their revenue on the domestic market. They experience slow growth, but they are still a minor actor in the IT industry, and they work on projects with limited size and budgets compared to the major actors in the defense industry. Teleplan Globe has a financial model with three sources of revenue; projects, licenses and maintenance. In projects for the Armed Forces, the defense finances the development and pays Teleplan for the hours put into the projects. This means that the Armed Forces own the finished products and therefore they get all licenses for free. Teleplan Globe owns the right to sell licenses abroad, and the management hopes that sale of licenses will contribute to their currently limited profits.

Source: Interviews and www.teleplanglobe.no (accessed 25.04.2014)

4.2.1 Methodology

One of the directors stated that the company originally used a lightweight methodology which utilized some “agile principles” but it was not a concrete methodology. In 2006 the company obtained their first experience with Scrum. They had reviewed literature and were familiar with other organizations which had used the methodology, but the personnel had no formal Scrum education. The first project was development of an electronic map and navigation system (FACNAV) which was developed to support the Special Forces’ cooperation with combat air-crafts. The methodology was experienced as an instant success and the informant who in 2006 worked as the project’s project manager stated that it worked great not only for project managers, but maybe particularly for users and the programmers. It took seven months from the software development started until the system was utilized in live operations in

Afghanistan. The former project manager believes this is unique and probably a world record. This was probably possible because the project was needed by a group of users that were prioritized and had representatives in the right places (Danielsen & Valaker, 2012). After the success Teleplan implemented Scrum as their standard methodology. They had their project managers trained and educated as certifies Scrum Masters and their programmers attained training in order to realize the methodology’s inherent potential. One of the directors stated that Teleplan Globe is very satisfied with the Scrum methodology and therefore alternative methods are rarely discussed. However, he emphasizes that they do not use Scrum as an iron bound process, it is modified to fit each project. Scope, timeframe and project complexity are considered important factors when they accommodate the methodology to specific projects. He further explains that the results are always paramount, and the methodology itself is always subordinate.

Framework

Although Scrum is the practiced methodology, it is especially designed to facilitate product development and certain additional tools are needed to manage projects. Teleplan Globe is a relatively small organization and they are determined to keep to administrative cost at a minimum and do not wish to outgrow its current boundaries and they do not have a heavy framework such as the Norwegian Armed Forces’ Prinsix. They use Agresso (an alternative to SAP/Oracle) as their ERP system and they have their own quality assurance (QA) system which manages the project plan and compares the expenditure with budgets. In addition they use Microsoft’s Project and the Office tools frequently.

The iron triangle

One of the directors expressed that Teleplan Globe are ready to accommodate customers’ needs in any way possible in order to reach desired results within suggested time frames. He explains that Teleplan Globe neither compromises on product quality nor time, and thus cost is their variable factor. This has resulted in an excellent reputation for product quality and their products are today widespread in the Norwegian Armed Forces. However, their profits are mentioned as unfortunately low. The current product owner of the NORBMS (formerly known as FACNAV) portfolio explained that even though the product is a huge success and has thousands of users, Teleplan Globe have still not earned back their investments, and although there is a potential in future sales of licenses, substantial numbers are needed and the seven licenses they have sold to Sweden this far do not contribute to their results on the bottom line.

4.2.2 Plan or Improvisation

A director of Teleplan Globe explained that they are conscious of risks in projects and that they endeavor to reduce risk in the early phases of their projects. Projects where risk cannot be reduced to acceptable levels are deliberately avoided. In order to assess risk, Teleplan Globe has to put together some elements of the project plan before tenders are issued. This early plan is also used for presentations for potential customers. If Teleplan Globe is awarded a contract, their self-developed QA-system (quality assurance) requires a project plan together with a budget before development can start. This system is also used for software project initiated internally in Teleplan. It was emphasized that a project plan and a budget is requirement in their QA system before any development can start. This means that the development needs to be broken down in work packages. The director stated that this plan is general and flexible, and that the planning process is paramount to the plan itself. The customer is often the Armed Forces, and although Teleplan Globe uses Scrum, customers often want plans on a more detailed level. One project manager points out that there are problems especially when cooperating with mercantile departments, since they are quite locked to the traditional waterfall process and demand a three year delivery plan including plans for the design phase including preliminary design review and critical design review. This is mentioned as a bothersome process which does not do anything good except that it is needed in Armed Forces rigid system.

One of the directors explained that the formal project plan is gradually replaced with Scrum-plans, given that it was accepted by the customer. In cases where the customer did not accept this transition, Teleplan Globe continued to use the formal project plans when they interacted with the customer, but they still replaced the project plan with scrum internally. A project manager explained that project managers give the developers timeframes, which is the basis for a Scrum program consisting of long term plans (six month sprints), which is again divided in sprints. The sprints usually lasts three to four weeks, and the general rule is that sprints should be fifteen working days, in other words three weeks usually and a bit longer in months which include national holidays. Sprints are in some cases longer, but never considerably shorter than three weeks. During these sprints, the development teams start their day with a fifteen minute planning meeting, the standup meeting, which usually starts at the same time every day and it is not postponed if somebody is not present. The Scrum master is responsible for this meeting and usually this role is often filled by a developer and not the product owner or project manager. A project manager estimates that this meeting is usually attended by eight to ten persons. Each person is invited to give a brief presentation of (1) What they accomplished yesterday (2) What they plan to do today and (3) What obstacles are impending the process. These meetings demand discipline in order to be kept short and intense. Lengthy discussions should be avoided during these meetings and full attention is requires by all attendants. One project manager at Teleplan Globe admits that although they intend to keep these meetings short and effective, the meetings have a tendency to drag out. A recurrence is that developers discuss design attributes in plenary, although these discussions should be held in smaller groups. One of the project managers admitted that although the standup meetings were supposed to last for almost exactly fifteen minutes, he often seized the opportunity to arrange brief meetings with his team directly after the standup meetings. This was an excellent opportunity to coordinate various activities which were not covered in the standup meeting.

4.2.3 Challenges and Critical Success Factors with the applied Methodology

Questions were asked about product backlogs, more specifically if it was ever problematic that Scrum has a tendency to prioritize the measures that are most appreciated by the customers? This was not seen as a problem, but something the project managers needed to bear in mind. There was always a lot of “invisible” work to be done “under the hood” which the users did not notice, but if this work was postponed, it could lead to greater problems in later stages. One of the project managers stated that such postponing was actually a problem in his portfolio. His projects portfolio consisted of products that were already in use, but development and maintenance was based upon a general agreement with the Armed Forces. The products were past the point where upgrades were recommended, but since the products still worked for the users, it was experienced as problematic to acquire the necessary funds for invisible, but important upgrades “under the hood”.

To be agile was experienced as challenging when cooperating with non-agile customers. Mercantile challenges have already been mentioned, and agility was said to be problematic when binding contracts had to be written. Mutual trust was mentioned as superior to any contracts when the goal was to develop good software. One of the project managers explained that he had experience with customers who believed more in contracts than in mutual trust, and his experience was that this led to less flexibility during the development phase, and that this had a negative effect on the product quality. Another project manager articulates that lack of trust leads to that the effects of potential synergies disappears. He felt that the armed forces had an unhealthy relationship to trust, and this had initially been a problem in the relationship with NDLO ICT. It had however improved a lot after years of cooperation, and it probably helped that he was well familiar with the actors “on the other side of the table”.

Documentation was also listed as a problem with the current methodology. One project manager expressed that the documentation was the first thing that suffered when the developers had a tight time limit. This was often problematic for the users and the customers, who did not get sufficient support documentation. However, it was also a problem for Teleplan. The project manager stated that this was very unfortunate in the long run. Better documentation could make activities like debugging less time-consuming and it would also make it easier to reuse developed components, which makes economic sense.

Proper documentation was seen as a potential source for written sharing of experience, but when it came to actual learning it was seen as less effective than oral knowledge transfer. One of the project managers mentioned a tradition he was pleased with which they called “Globe’s Informal Pulpit”. It was a gathering they used to have on Fridays, where the employees met for a few hours, made waffles and shared experiences. The lectures were usually results of requests from the management who asked specific project teams to share their experiences.

The director admitted that these forums had a predominance of lectures on “cool stuff”. These lectures were usually held by Teleplan Globe employees, but they sometimes invited external speakers.

5 Cross-case discussion

The cross case discussion uses the same headings as the previous chapter, before the research questions are reintroduced and answered based upon the somewhat broader discussion presented under each heading.

5.1 Methodologies

Teleplan Globe has incorporated Scrum as its standard methodology for software development, but the methodology varies with scope, timeframe and project complexity. In other words the methodology is adjusted to fit each individual project. User representatives and the Scrum team work together to establish a product backlog, a “job list” which states which future tasks that have been identified. During the development phase, work is divided in sprints which results in potentially shippable products. Although the products are rarely shipped before a project is completed, the developers present these unfinished products to user representatives and together they update the product backlog, and decide which tasks from the product backlog should be prioritized in the succeeding sprint. Tasks get prioritized based upon an assessment of which measures that would deliver most added value to the users combined with an estimate of how much time each of these measures will cost. This assortment of tasks is referred to as the “sprint backlog”. In other words, Teleplan Globe does not integrate all developed components at the end of a project, but continuously. This growing functionality in series of increments indicates incremental development as described in section 2.1.3. The fact that they do not follow a detailed plan which was stated in the early phases of the development rules out a linear process (described in section 2.1.2). After each sprint the Scrum team stops to evaluate the result, and this evaluation is used to determine the continuation of the development process. When it is necessary they repeat certain parts from the last sprint cycle. This indicates iterative development (see section 2.1.4) because they “stop, evaluate and rework” until they have reached the desired outcome. Scrum seems to be a methodology which is designed to combine iterative and incremental development. The sprint periods are not only increments which results in potentially shippable products, they also act as basis for altering the plans when it is considered to be beneficial for the project. The author presents Figure 5-1, which he believes gives a general overview of Teleplan Globe’s incremental *and* iterative methodology; Scrum. The incremental model (Figure 2-3) was used as a basis and a loop was added. This loop allows plans to be altered when it is assumed to be beneficial.

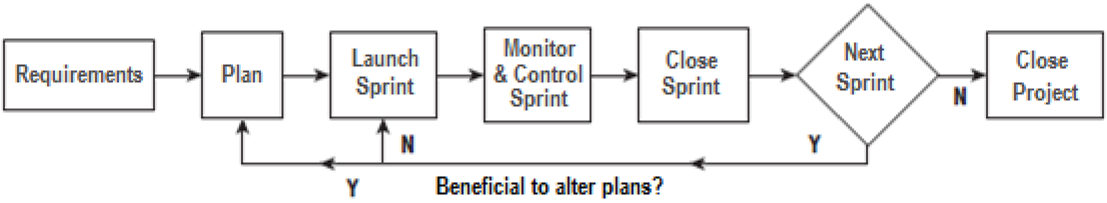

Figure 5-1 Teleplan Globe's Methodology

Source: Modified by author, based on Wysocki (2009)

The Norwegian Armed Forces appear to have a rigid, standardized framework for project management, which includes, but is not limited to, ICT or software development projects. Interviews done at NDLO ICT indicated that the framework has a life-cycle focus and does not focus on product development in particular. It describes a process from the conception of a project idea until a product eventually is out-phased or replaced. Several informants reported that Prinsix is too general to actually restrict the actual product development process (in this case software development), and that in practice, the vendors' own incorporated methodologies are used for software development. Although this might imply that the Armed Forces do not have a methodology for software development, the framework demands certain plans, documents, procedures etc. and the sum of control elements seems substantial. Therefore it does not seem fair to indicate that the Norwegian Armed Forces fully accepts vendors' software development methodologies as they are; the vendors need to adjust their methodologies in order to accommodate the Prinsix framework.

Several interviewees from Teleplan Globe expressed that they saw it as challenging to work with someone when they were agile and flexible, but had to cooperate with someone who was rigid and inflexible. Some of the Teleplan employees expressed that it was difficult to develop the best software possible when the development process was driven by a contract rather than mutual collaboration between users and developers. It was claimed that this removed the vendor's incentives for early delivery or cost savings and it was emphasized that mutual trust is a critical success factor to establish a sound relationship – which again is a prerequisite for successful software development. These statements align with the Agile Manifesto which states that the Agile Alliance values "customer collaboration over contract negotiation" (Beck et al., 2001).

A project manager from Teleplan Globe who worked on a protracted project reported that although they experienced some problems cooperating with NDLO ICT initially, this relationship had gradually improved over the years as the actors learned to understand each other. He repeatedly pointed out that although the relationship was difficult at first, he currently experienced an outstanding cooperation with his counterpart from the Armed Forces. This indicates that the Armed Forces framework does not encourage agile development, but that it can support agile methodologies given that project managers representing the Armed Forces and their counterparts from the industry mutually invest in the relationship. Based upon the impression from the interviews the Figure 5-2 is presented. Two way arrows have been drawn in order to show that the process is not the traditional waterfall model as presented in Figure 2-5, but the iterative waterfall model (presented in Figure 2-7). However, there are certain mercantile impediments which prevent some alterations of requirements and plans. Although these are illustrated by one way arrows, some minor changes can be done to initial requirements and plans, but major alterations might cause termination of a project before completion (further explained in section 4.1.2). The "vendor's development process" box might be replaced with figure Figure 5-1 in order to illustrate the cooperation between NDLO ICT and Teleplan Globe.

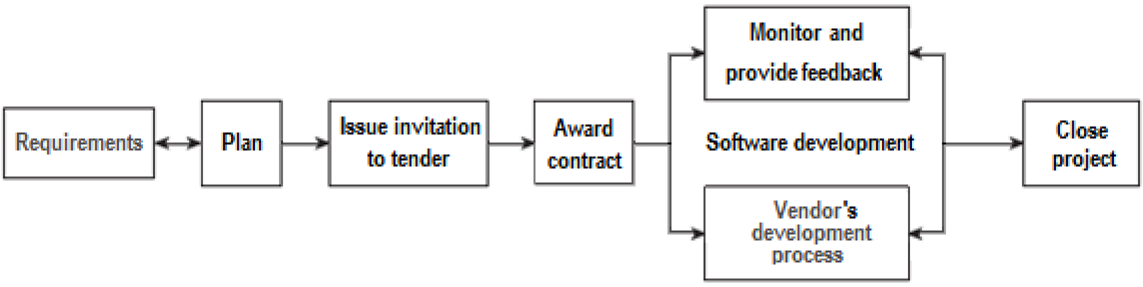
Requirements
Plan
Issue invitation to tender
Award contract
Monitor and provide feedback
Software development
Vendor's development process
Close project

Figure 5-2 NDLO ICT's Methodology

Source: Modified by author, based on Wysocki (2009)

The iron triangle

The iron triangle was particularly often mentioned during the interview sessions at NDLO ICT and it was easy to get the impression that this was a set of constraints that the employees had accepted, either based upon experience or as a result of Prinsix education. One informant reported that he saw the product quality as the variable, while two informants saw it as fixed.

The researcher believes that this discrepancy relates to whether they understood product quality as an absolute term (independent of the surroundings), or as relative to the best solutions available when the project was completed. Cost was generally seen as fixed, although some reserves were available and more funding could be granted if necessary. Time was seen as the most variable factor. The informants from Teleplan Globe were also familiar with the iron triangle, but they had more consistent answers. They saw the product quality and time as fixed constraints, but the cost was described as variable. In retrospect the researcher suspects that what the two organizations seem to have a slightly different understanding of the factors within the triangle, and they should probably have been accurately defined before the interviews. The NDLO ICT informants' seem to understand cost as the total cost of a project, while Teleplan Globe seem to think of cost as the number of hours spent multiplied with their cost of man hours. The gathered results regarding the iron triangle might be inaccurate as they are highly dependent of the interviewees' personal interpretations and although the researcher expected that the iron triangle would be able to explain the differences between the two methodologies, the empirical results does not support this and thus the significance of the triangle has been considerably toned town compared to the researcher's original intention.

5.2 Plan or Improvisation

The Norwegian Armed Forces apparently do not leave much to chance when they work on projects, and both the planning process and the resulting plans seem to be of high value. It seems fair to consider the planning phase as strictly regulated by the overarching Prinsix framework. However, the informants expressed that plans were not experienced as iron bound as they could be altered retrospectively, but that it demanded a certain procedure, and probably for good reasons. The plans are the results of a comprehensive process where stakeholders including competence authorities, mercantile/legal representatives have been represented, and it would be unfortunate if these plans could be altered without a defined protocol.

Teleplan Globe has a similar approach to planning in the initial phases of projects. Teleplan initiates (or refuses) projects based on project plans, budgets and a methodical processes of risk assessment. It was expressed that this had to do with necessary presentations for potential customers during tender processes and it was seemingly a mandatory starting point for projects, at least in projects inquired by governmental organizations. Teleplan Globe also has some software projects initiated internally, which they develop and later offer to customers, but their QA system still requires a project plan and a budget before development can be initiated. It therefore seems fair to claim that Teleplan Globe starts their development projects traditionally. This preface to software development seems to require a certain amount of precision and structure. When Teleplan Globe has been awarded a contract and a project moves into the development phase, a gradual shift occurs. Scrum processes gradually replace traditional project plans, but although agile is often associated with less focus on planning and more focus on development of good software, the researcher sees a glimpse of structure underneath the surface. Teleplan Globe develops general sprint plans for six months, and these periods are again divided in three-week sprints, which results in potentially shippable products (incremental development). Every day starts with a structured coordination meeting where the development team including product owners and project managers are present. Activities are coordinated and necessary meetings and discussions are scheduled. This strikes the researcher as similar planning activities that were observed at NDLO ICT, although a lot of this coordination was done using Outlook Calendar and intraday planning seemed to be less widespread. One of the reasons for this difference is perhaps that Teleplan Globe employees are usually dedicated to one single project, and most of the necessary participants are located in-house. All informants from NDLO ICT had their portfolios of projects, and many of these projects demanded coordination with external actors. Intraday planning appears to be problematic when people work on several parallel projects and the actors are not co-located.

Teleplan Globe is a part of the agile community and they seem to agree that the planning process is more important than the actual plans. However, it does not seem fair to indicate that the Norwegian Armed Forces see this the other way round; that the plans are more important than the planning process. – For what are plans without a successful planning process? Does anyone see the plans as paramount to the planning processes? The researcher believes that this variation has to do with project size or more specifically the number of involved personnel; while agile software development teams can fit within a conference room and communicate effectively on a day to day basis, this seems difficult for large organizations with a broad array of geographically scattered stakeholders.

The Norwegian Armed Forces regard the resulting plans as important, while Teleplan does not consider the plans as equally important. The researcher believes that this different view on the plans has to do with the number of personnel involved in the projects. While Teleplan Globe's development team can plan exercise intraday planning, this seems to be difficult for NDLO ICT where numerous personnel are involved, where people are not dedicated to certain projects but have portfolios of projects and where the personnel are often geographically scattered. This probably increases the importance of a long term plans including plans for future meeting activities.

5.3 Challenges and Critical Success Factors with the applied Methodologies

The informants from Teleplan Globe stated that their goal is to develop as good software as possible for its users and customers, but that they are dependent on mutual flexibility to achieve this. Flexibility requires trust, and lack of trust shifts the focus from developing the software to fulfilling contractual terms. They tell a story of increasing trust over the years they have worked for the Norwegian Armed Forces and are currently enjoying a healthy cooperation, especially with the project managers from NDLO ICT.

One of the project managers at NDLO ICT said that the fact that people were not dedicated to projects was unfortunate and a common cause of delays in projects. Some specific fields of special competence are particularly vulnerable and often one-man deep. In certain situations projects were delayed until these key individuals were able to allocate time for certain projects. He suggested that specialists should be dedicated to certain projects, rather than letting prioritizing be up to each of the key individuals. Teleplan Globe usually has their developers dedicated full time to projects, which seems beneficial, but this is probably difficult to achieve for NDLO. Teleplan Globe has only 70 people and has a limited amount of ongoing projects, while the NDLO have around 330 employees of a much broader composition of personnel and continuously have a portfolio consisting of more than 40 projects. Some of these employees have unique competence, and some disciplines consist of only one person. This means that the Armed Forces have access to personnel with very specific domain knowledge, but since they act as key individuals in several projects, the access to these key individuals can soon become bottlenecks, ultimately hampering the progress of the projects. This was also supported by the FFI report on the NDLO which recommended a reinforcement of certain critical specialist environments (Gulichsen et al., 2012).

Both NDLO ICT and Teleplan Globe reported problems regarding documentation, and the interviews indicated that the former produced a lot of documentation, but not necessarily enough of certain types of documentation such as documents where experience and lessons learned were shared. The latter experienced that the documentation was the first thing to suffer when the time limits were tight. This was experienced as unfortunate, as it made it comprehensive to upgrade existing software and difficult to reuse developed components. Knowledge transfer was considered as particularly challenging in the NDLO, and in practice dependent of personal networks. It was expressed that there were no established credible systems which ensured knowledge transfer between projects. Teleplan Globe has considerably fewer individuals and although they did not have a structural framework which supported knowledge sharing, they regularly had informal meetings where they shared experience.

5.4 Summary of the cross case discussion

5.4.1 Methodologies

What are the characteristics of the methodology applied by the Norwegian Armed Forces and of the methodologies applied by Teleplan Globe?

Teleplan Globe applies a traditional methodology during the early project phases, as this is required in order to be considered in the competitive bidding process before contracts are awarded. In the subsequent development, the traditional methodology is replaced by a methodology for software development called Scrum. The methodology is adjusted to fit with each individual project, depending on scope, timeframe and project complexity. Scrum is in principle a standardized incremental and iterative methodology.

The NDLO ICT uses Prinsix, which can be argued to be a framework rather than a methodology. Prinsix standardizes the Norwegian Armed Forces’ investment processes and it is not limited to ICT projects or product development. It is often referred to as a waterfall approach, which indicates linear development, but although altering of plans is difficult, it is not impossible. An informant described the process as a waterfall approach with arrows pointing in both directions. In principle, this indicates a process where plans can be altered based upon experience, which indicates that Prinsix can be iterative. The conclusion is that the Prinsix process is linear, but that iterative development is difficult and not impossible. However, this difficulty is probably not only caused by the framework, but also by the Norwegian laws for competition. These laws seem to restrict the possibility to alter the requirements and with that the plans once a contract has been awarded to a vendor. Although this is often understood as a problem with the Prinsix framework, at least some of these restrictions are believed to be caused by the national legal restrictions. Therefore, the researcher understands the NDLO ICT methodology as partly linear and partly iterative. Although the Norwegian Armed Forces have a legacy for integration in the later project stages, the author believes that Prinsix also can support incremental build if it is encouraged by the important stakeholders. However, it should be mentioned that no reliable data has been found which suggests that the Prinsix framework encourages or prevents incremental development.

5.4.2 Plan or Improvisation

To what extent are projects pre-planned? Are plans followed mechanically or used as a basis for continuous modifications as projects progress?

The research question seems best answered with that while the NDLO ICT places a massive effort into long term plans which concerns the whole project life cycle, although an informant expressed that the timeline was unreliable beyond three months' time. In other words the plans are written early, but the timeline is increasingly unreliable the further away certain planned activities are in time. Although the timeline was uncertain, the plan was to some extent followed mechanically. A NDLO ICT informant stated that the rigidness varied with project type. The vast projects where NDLO ICT collaborated with the large actors in the defense industry were reported as particularly rigid, and only open to "cosmetic changes", while some smaller projects had plans which were less rigid of nature.

Teleplan Globe apparently invests less energy in long-term plans, and the early project plans are meant as points of departure, rather than rigid plans. If this is accepted the initial plans are replaced with general six-months Scrum plans, and the more detailed sprint plans which define activities for a sprint period (usually three weeks). The researcher got the impression that long term plans were general and uncertain, while short term plans were followed more mechanically. In other works, the six-month plans were uncertain, but the sprint plans were more reliable. Intraday plans were followed more or less mechanically.

5.4.3 Challenges and Critical Success Factors with the applied Methodologies

What are the experienced challenges and critical success factors with the applied methodologies?

Informants from both NDLO ICT and Teleplan experienced documentation as both challenging and a critical success factor. Some of the NDLO ICT informants reported that they used too much time on laborious documentation which they did not regard as important, as they felt it is was written only to reach milestones and did not serve a practical purpose. Teleplan Globe informants expressed that they took too lightly upon documentation in general, as documentation was the first thing that suffered when they worked with tight time schedules, and the researcher got the impression that tight time schedules was rather the rule than the exception for Teleplan Globe.

Informants from NDLO reported that they saw organizational learning as particularly challenging and that they regarded it as an area for improvement. Although they used the Prinsix framework, which actually demands a systematic gathering of lessons learned and experiences in a report, the termination reports were difficult to access after completion and therefore rarely used. This was seen as unfortunate, as it limited the potential for knowledge transfer especially across projects. The NDLO ICT had also identified a potential for better systems for knowledge transfer, but they were fewer individuals and had established a routine with informal monthly meetings in order to improve the internal knowledge transfer.

The NDLO ICT informants regarded the access to personnel as both challenging and a critical factor for success. The project managers were often dependent on work packages from certain key individuals and the lack of access to these individuals was regarded as a common source of delays.

Table 5-1 is a brief summary of the cross case analyses.

	NDLO ICT	Teleplan Globe
Methodology	The Prinsix framework overarches the methodology, but does not mandate a specific development methodology.	Scrum, tailored to fit each project based on scope, timeframe and project complexity.
View on the iron triangle	Product quality and cost is fixed. The timeframe varies.	Fixed timeframe and product quality. Cost is variable.
Plan or Improvisation	Both the planning process and the resulting plans are important.	Planning more important than the plans.
Stakeholders and user involvement	Elaborate analyses of stakeholders. Users are important stakeholders among others. The Department of Defense is the supreme stakeholder.	Fewer stakeholders to interact with, POCs rather than individuals. Users are considered supreme stakeholders.
Challenges/Critical success factors	Timely access to key personnel. Organizational learning and knowledge transfer across project teams.	Mutual trust between the customer's representatives and the team of developers.

Table 5-1 Summary table of the cross case analyses

6 Recommendations and conclusions

6.1 Conclusions

When the first literature on agile development was read it seemed obvious that agile development was superior to traditional, rigid development, and this was reflected in the original problem formulation. Who could possibly argue against the principles from the agile manifesto, they actually seemed self-evident. A list of numerous recommendations based upon the study seemed within reach. However, after several rounds of interviews with individuals working both “traditionally” and “agile”, the “revolutionary slogans” had to be put on the shelf and the problem formulation and research questions had to be rewritten. Agile seems to make sense where people are co-located, when they work together in small dedicated teams and where development of good software is declared the number one priority. However, agile processes seem less tempting in organizations where large groups work together, where they work on portfolios of projects rather than focusing on one project at a time or where the involved actors are geographically scattered. The researcher believes that agile methodology is not for everyone, and it would be impossible for the Norwegian Armed Forces to adapt agile methodology as a replacement for Prinsix. When that is stated, the researcher believes that agile methodologies have certain advantages when it comes to rapid deliveries of software built on users’ premises for a reasonable price. Rapid deliveries and “low” cost is perhaps a result of the limited amounts of personnel involved and a lower amount of labor-intensive procedures than required with traditional methodologies. The user friendliness might be a result of the agile methodologies’ emphasis on close collaboration with users. Agile development seems to have an advantage when it comes to delivering fast value for users and customers, while large governmental organizations such as the Norwegian Armed Forces might be superior when it comes to carrying the heavy workloads. Some of the Armed Forces’ procurement projects last for more than 20 years, involve hundreds of people and budgets of billions of dollars. In such projects quick fixes and rapid value is probably not considered a top priority.

From the Norwegian Armed Forces’ perspective, projects seem to go through the same processual framework regardless of project nature. The framework has a life-cycle focus, and no specific focus on product development. This framework might seem meticulous, but the research results indicated that although it standardizes the organization's project management, it does not mandate a specific methodology for product development. The Norwegian Armed Forces apparently do not have a dogmatic relationship to software development methodology and utilize their vendors' practices during this phase. Teleplan Globe appears to have a stronger determination regarding their approach to software methodology. Although they use project plans and budgets in the early phases, they gradually replace the traditional approaches with Scrum, which is an incremental and iterative methodology. If the customer objects to this, Teleplan Globe runs two parallel processes. The interface towards the customer depends on the customer's preferred methodology, but they still use Scrum internally to develop software.

The agile community often states that they see the planning process as more important than the actual plans, and it may be tempting to sign the manifesto. However, plans are the results of planning processes and it seems difficult to imagine that anyone sees plans as more important than the process. The research question seems best answered arguing that while both organizations see the planning process as extremely valuable; the Norwegian Armed Forces regard the resulting plans as important, while Teleplan regard plans of secondary to the process. The researcher believes that this variance can be related to the number of involved personnel and project scope.

Teleplan Globe interacts closely with the customer, and the informants expressed explicitly that the users were their most important stakeholders. The interviews from NDLO ICT indicated that although users were important stakeholders, they were among many other stakeholders the project groups had to take into consideration. NDLO ICT saw the Department of Defense as the most important stakeholder, but also the Contract Department and the Security Department were regarded as especially important stakeholders. The researcher sees this as a variation which can be tracked back to the different organizations scopes of work. While Teleplan are well adapted to develop software to users, the Armed Forces have a life-cycle perspective, and while the user certainly is important with a life-cycle perspective, there are several other stakeholders which the NDLO ICT needs to consider as equally or in some cases even more important.

Both organizations regarded documentation as problematic. While the Teleplan Globe employees felt that they produced insufficient documentation, interviews from NDLO ICT indicated that the level of documentation in some areas were extreme but that it was insufficient in areas regarding experience and lessons learned. Although documentation can lead to organizational learning, there is no direct connection. The NDLO ICT writes comprehensive reports, but interviews and a report suggest that the underlying potential in these reports is not utilized, as they are difficult to find and there is no established database for these reports.

The thesis was introduced with a question which asked whether the Norwegian military acquisition machinery was in need of some drops of oil or comprehensive reengineering when it comes to acquiring software. Although the researcher admits that he had a bias which suggested an extensive reengineering process, this bias has been repeatedly challenged during the research process. The military bureaucracy might appear vast and inflexible, but the practice seems to be built on experience and it probably is no coincidence that the foreign military organizations known to the researcher is organized in similar bureaucracies. These organizations are perceived as inflexible, comprehensive and incredibly expensive, but credible alternatives have not been introduced yet. Lightweight methodologies such as agile appear to have an advantage when it comes to software development, but software development is only a minor fraction of the military mission portfolio. The Armed Forces need to accommodate a broad array of tasks, which might justify the heavyweight bureaucracy. All in all, this indicates that military machinery needs some drops of oil rather than an extensive reengineering process.

The comparison of Teleplan Globe and the NDLO ICT in retrospect feels like comparing a single gear with a vast machine. Teleplan Globe is a niche actor in the IT industry with a few specific fields of expertise within software development. In contrast, the NDLO ICT has a wide spectrum of tasks where software development constitutes only a fraction of their portfolio, and most projects are combinations of software, hardware and infrastructure. It is therefore doubtful that experience and best practices established in one of the units can be directly transferred to the other; the organizations are fundamentally different.

6.2 Suggested areas for improvement

An interesting discovery was that the Norwegian Armed Forces actually seem to have good routines for systematic documentation of experiences and lessons learned, but apparently the resulting documents are perceived as difficult to access and the documents are rarely used after project termination. Some projects are likely to be troublesome, and while some failures might be inevitable, the military project organization needs to learn from these failures to avoid repeating the same mistakes. This requires effective knowledge management, and the researcher would like to suggest a project which examines current practices for knowledge sharing and tries to define a set of best practices. A NDLO ICT project manager suggested a database with a helpful searching function. This could probably contribute to better knowledge management. However, even the best structural systems will not work if a culture for knowledge sharing is absent, and although the researcher cannot claim that there are problems regarding the cultural component, it cannot be ruled out and the cultural aspect should therefore also be thoroughly examined.

Teleplan Globe informants reported that they saw it as challenging to work with someone who was non-agile. Mercantile challenges and contracts were listed as particularly challenging. The researcher would therefore suggest that problems between the NDLO and their agile vendors are systematically investigated, before a review of the current practices is conducted. The interviews reported that the mutual cooperation usually improved when project managers from both organizations were acquainted and it seems important to establish practices for cooperation which does not depend on personal relationships.

6.3 Strengths and limitations of the thesis

Although the case study does not provide “hard and reliable” data the researcher believes it provides “rich and deep” (see Table 3-2) material which might have value beyond the units of observation. However, the generalizability is a major limitation with this thesis and it is at best questionable. It is a descriptive case study which examines two very different organizations and the sample of informants from both organizations is of very limited size. Therefore the thesis conclusions might not be applicable for generalization as there is a possibility that the sample from each organization was unique and that the informants provided an erroneous picture of their respective organizations. There is also a possibility that the selected cases are unique and that other units of observation would have led to different conclusions. This is a common problem with case studies which rely on qualitative data. Particularly the five person sample from Teleplan is unfortunate. It would have been very beneficial to supplement the conducted interviews with interviews of personnel from other similar agile vendors. – A broader selection of informants generally leads to better reliability. The Department of Defense’s perspective would have been a great supplement to the performed interviews, as they were referred to as the customer for military procurement projects. Their answers probably would have been interesting and a great supplement to the conducted interviews. Several attempts were made to gain access and the correct key individuals were identified. Unfortunately, written inquiries were sent, but not answered, and the researcher was unable to reach the correct individuals by phone. The researcher sees in retrospect that he should have started this process earlier and his used personal network together with the official channels in order to gain access.

Another limitation is the research method. Some of the research questions asked questions which proposed the informants to list e.g. challenges and critical success factors. The interviews did not lead to any complete sets of such factors. The researcher expects that other research methods, such as a protracted participant observation, a survey or especially a study of documents from completed projects would have been great supplement in order to provide a more complete set of answers to the research questions.

If a key strength should be identified it would have to be that although the informants were few, they were experienced in their work, had diverse backgrounds and apparently had thought through the interview questions prior to the interviews.

6.4 Suggested areas for future research

In the literature review, software with great criticality was regarded as a home ground for traditional development. Although the research questions did not specifically ask about criticality, a question regarding methodology and software with high criticality was written in the interview guide in order to understand the methodologies. The answers did not provide any consistent results. The researcher would like to suggest that the connection between the development methodology and the suitability for development of software with high criticality is examined.

Especially informants from the NDLO ICT indicated the requirements were the foundation of the project plan and that it was crucial that requirements were written in a way which allowed the right amount of flexibility throughout projects. A study of how the writing of initial requirements can affect progress and product quality could provide valuable insights.

The interviews revealed that the informants experienced certain challenges with agile development and the Prinsix framework. Mercantile challenges were mentioned explicitly and the interface between the customer and vendors was seen as problematic. It would be interesting to perform a study on the problems between “lightweight” vendors and customers which apply “heavyweight” methods/frameworks.

7 References

Alleman, G. B. (2002). Is There an Underlying Theory of Software Project Management?(A critique of the transformational and normative views of project management) Version 2.4, 9.1. 2002.

Ambler, S. W. (2001). When Does(n't) Agile Modeling Make Sense? Retrieved 10.12, 2013, from http://www.agilemodeling.com/essays/whenDoesAMWork.htm

Ambler, S. W. (2005). Disciplined Agile Software Development: Definition. Retrieved 03.03.2014, 2014, from http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm

Ambler, S. W. (2006). Examining the Agile Manifesto. Retrieved 10.02, 2014, from http://www.ambysoft.com/essays/agileManifesto.html#Values

Atkinson, R. (1999). Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International Journal of Project Management, 17*(6), 337-342.

Baxter, P., & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. *The qualitative report, 13*(4), 544-559.

Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved 02.02, 2014, from http://agilemanifesto.org/

Bellomo, S. (2011). A Closer Look at 804: A Summary of Considerations for DoD Program Managers.

Boehm, B. (1996). Anchoring the software process. *Software, IEEE, 13*(4), 73-82.

Boehm, B. (2002). Get ready for agile methods, with care. *Computer, 35*(1), 64-69.

Boehm, B. (2006). *A view of 20th and 21st century software engineering*. Paper presented at the Proceedings of the 28th international conference on Software engineering, Shanghai, China.

Boehm, B., & Turner, R. (2003). *Balancing agility and discipline: A guide for the perplexed*: Addison-Wesley Professional.

Brooks Jr, F. P. (1987). No silver bullet-essence and accidents of software engineering. *IEEE computer, 20*(4), 10-19.

Bryman, A. (1984). The debate about quantitative and qualitative research: a question of method or epistemology? *British Journal of Sociology*, 75-92.

Bryman, A. (2003). *Quantity and quality in social research*: Routledge.

Bryman, A., & Bell, E. (2011). *Business Research Methods 3rd Edition*: Oxford University Press.

Buresh, D. (2008). Customer Satisfaction and Agile Methods. *IEEE Reliability Society*.

Cockburn, A. (2000). Selecting a project's methodology. *Software, IEEE, 17*(4), 64-71.

Cockburn, A. (2008). Using both incremental and iterative development. *STSC CrossTalk (USAF Software Technology Support Center), 21*(5), 27-30.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer, 34*(11), 131-133.

Constantine, L. (2001). Methodological Agility. *Software Development*(2001), 67-69.

Creswell, J. W. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*: Sage.

Daltveit, E., Ydstebø, P., & Geiner, J. F. (2010). Trender i militære operasjoner (pp. 78 s.). Kjeller: Forsvarets forskningsinstitutt.

Danielsen, T., & Valaker, S. (2012). Teknologisk inovasjon med fart og retning i spesialstyrkene: en antropologisk studie (pp. 101). Kjeller: Forsvarets forskningsinstitutt (FFI).

DeBellis, M., & Haapala, C. (1995). User-centric software engineering. *IEEE Expert, 10*(1), 34-41.

Gerring, J., & Thomas, C. W. (2011). Qualitative versus Quantitative Methods: A Question of Comparability. In B. Badie , D. Berg-Schlosser & L. Morlino (Eds.), *International Encyclopedia of Political Science*: Sage Publications.

Guba, E. G., & Lincoln, Y. S. (1994). Competing paradigms in qualitative research. *Handbook of qualitative research, 2*, 105-117.

Gulichsen, S., Strans, K. R., & Steder, F. B. (2012). Forsvarets investeringsprosess - flaskehalsanalyse og kompetansekartlegging i FLO. Kjeller: Forsvarets forskningsinstitutt (FFI).

Highsmith, J. (2001). History: The agile manifesto. Retrieved 12.12, 2013, from http://www.agilemanifesto.org/history.html

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer, 34*(9), 120-127.

Kanselaar, G. (2002). Constructivism and socio-constructivism. *Article published on July, 16*, 2002.

Krauss, S. E. (2005). Research paradigms and meaning making: A primer. *The qualitative report, 10*(4), 758-770.

Kruchten, P. (2001). From Waterfall to Iterative Development—A Challenging Transition for Project Managers. *Rational Edge, Rational Software*.

Larman, C. (2004). *Agile and iterative development: a manager's guide*: Addison-Wesley Professional.

Mertens, D. M. (1998). Research methods in education and psychology: Integrating diversity with quantitative & qualitative approaches.

Monson-Haefel, R. (2009). *97 things every software architect should know: collective wisdom from the experts*: O'Reilly.

Morgan, G. (2006). *Images of organization*. Thousand Oaks, Calif.: Sage.

Paulk, M. C. (2002). Agile methodologies and process discipline. *Institute for Software Research*, 3.

Rakitin, S. (2001). Manifesto elicits cynicism. *IEEE computer, 34*(12), 4.

Royce, W. W. (1970). *Managing the development of large software systems*. Paper presented at the proceedings of IEEE WESCON.

Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes, 35*(3), 8-13.

Sapolsky, H. M., Gholz, E., & Kaufman, A. (1999). Security lessons from the cold war. *Foreign Aff., 78*, 77.

Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research methods for business students*. Essex: Pearson Education Limited.

Schwaber, K. (2004). *Agile project management with Scrum*: O'Reilly Media, Inc.

Tellis, W. (1997). Application of a case study methodology. *The qualitative report, 3*(3), 1-17.

Trochim, W. M. K. (2006). Research Methods Knowledge Base, 2nd Edition. Retrieved 02.02, 2014, from http://www.socialresearchmethods.net/kb/

Woodside, A. G., & Wilson, E. J. (2003). Case study research methods for theory building. *Journal of Business & Industrial Marketing, 18*(6/7), 493-508.

Wysocki, R. K. (2009). *Effective Project Management: Traditional, Agile, Extreme* (5th ed.). Indianapolis, Indiana: Wiley Publishing, Inc.

Yin, R. K. (2014). *Case study research: design and methods* (Fifth edition ed.). Los Angeles, Calif.: SAGE.

Appendices

Appendix A – The Agile Manifesto

Manifesto for Agile Software Development:

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

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.

Principles behind the Agile Manifesto:

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.

Source: Beck et al. (2001)(numbers added by author).

Appendix B – Participant Information Sheet

Background

I have contacted you because you are working or have worked with software development, either within or in cooperation with the Norwegian Armed Forces. According to research ethical guidelines, it is important that participants in a research projects have the opportunity to familiarize themselves with the research project's content, and what consequences it might have to contribute to the study. The purpose of this letter is to give you an opportunity to understand the thesis' purpose, for what use it is intended and give you time to think about whether there is information related to the interviews that you think may be sensitive or that you do not want to share with others. The thesis is unclassified, so any classified information that I gain access to will not be included in the final report. If we touch into material you believe is classified or sensitive, please make me aware of this and I will respect your concerns.

About the researcher

I am writing my final thesis as part of NTNU's Master of Technology Management and it will be completed by the summer of 2014. The term "agile" is central in the thesis and it asks the questions what methodologies are suitable for software development for the Norwegian Armed Forces. The thesis is a descriptive multi case study that examines NDLO ICT and Teleplan. I have written a theoretical review and will compare this theory with the collected data from interviews and reports and see how methodologies for software development are exercised in practice. In order to get data material with satisfying quality, personnel of various categories (age, experience, position etc.) will be interviewed. A total of 10 to 15 interviews will be collected. Written reports will also be used to retrieve data.

The interview guide is written with basis in the following research questions:

1. What are the characteristics of the applied methodology?
2. To what extent are projects pre-planned? Are plans followed mechanically or used as a basis for continuous modifications as projects progress?
3. Which stakeholders have influence in the projects? Do users hold an exceptional position? [Questions regarding stakeholder involvement were asked, but not included in the final report].
4. What are the experienced challenges and critical success factors with the applied methodology?

Practical information regarding the interviews

The interviews will be conducted as formal interviews. A semi-structured interview guide will be sent out in advance. The interviews will last for approximately 40 minutes and I hope you can find time to prioritize this in a hectic workday. I would prefer to use a voice recorder during the interviews and please let me know if you have any concerns with this. If there are certain topics you would like to discuss without voice recording, please inform me and I will pause the recorder temporarily.

Security

I consider the recorded sound files as unclassified, but they will be within my possession at all times and taken good care of. The recorder will first be turned on after we have been introduced to each other and your name will not be mentioned during recording in order to conserve your anonymity. I will, however, maintain a list where I connect your name to the specific interview, in case I will seek your permission for quotes, ask you some clarifying questions on a later stage, or give you the opportunity to read through the interview. All recordings will be stored on an external hard drive, together with all transcribed material will be stored. Both audio files and text files will be deleted after the thesis has been evaluated, a grade is awarded and the final appeal deadline is exceeded. When the thesis is completed it will be censored by an external examiner appointed by NTNU. If the thesis achieves grade B or better, it will be made available through NTNU's archive for educational theses.

Contact information:

Role:	**Name:**	**Contact info:**
Academic supervisor NTNU	Dr. Arve Pettersen	Phone: +47 95293138 arve.pettersen@hist.no
Academic supervisor FFI	Frode Rutledal	Phone: +47 99436380 frode.rutledal@ffi.no
Responsible for NTNU’s MTM program	Dr. Endre Sjøvold	Phone: +47 41300300 endre.sjovold@iot.ntnu.no

Please contact me with any concerns, if would like any further information or if you do not want to contribute to the study. A semi-structured interview guide will be sent out together with this information sheet.

Best regards,

Thomas Nymo Skogvold

Phone: +47 92049402

Email: tskogvold@gmail.com

Appendix C – The Interview Guide

Introduction

1. Please describe your role in the organization. Education/background/number of years in this position and please list your working experience.
2. Which projects are you currently working on? How is your time divided between different projects?
3. Can you please describe a "typical" software development project for your organization? Please describe number of involved personnel, type of project, budget, duration… –does this differ with project's phase?
4. With which methodologies do you have personal experience? If more than one, please explain the distribution between the different methodologies.
5. How did you gain experience with these methodologies?

Plan or improvisation

1. Describe the planning stages of a typical software project in your organization.
2. To what extent do the plans change during the project? Is the plan static or dynamic? Please explain.
3. Are there any other components that are particularly difficult to plan? Requirements, technical specifications, technology or any other elements?
4. When the development of new software starts, to what extent have you understood the final product? To what extent does your understanding of the product change during development?
5. How often do you meet in workgroups during a project? Who is represented in these meetings? What kinds of topics are discussed and what kind of decisions are made in these meetings?
6. How are the software developer teams organized? Please elaborate on location, size, etc.
7. How far in advance do you know of the specific tasks that are to be completed either by yourself or your colleagues in a project?

Methodology

1. Which tools/methodologies/models do you use for the management of a software development project?
2. To what extent are the projects linear, incremental or iterative?
3. Why do you use this/these methodologies?
4. How do the different methodologies you have experience with fit with the nature/business of the Norwegian Armed Forces? Have you run into any specific problems when working with the Norwegian Armed Forces?
5. Are there any regular problems that are experienced repeatedly? What type of regular problems?
6. In which project phases is it common that such problems occur/are revealed?
7. What is needed to terminate a project? In which phases can project termination occur?
8. How do you measure the degree of success of a project (internally/externally)? Is it measured continuously, or only project completion?
9. With reference to the iron triangle of time, cost and quality, are any factors more fixed or variable than others? Why do you believe this is this so?
10. How do you evaluate the development risk throughout the different phases of a project?

User Involvement

[Comment: early versions of the thesis dealt with stakeholders and the role of end users during software development, but this was removed in later versions. Although these questions are not directly relevant for the final thesis, the interview guide is reproduces as it was presented to the participants].

1. Which stakeholders are taken into consideration in a project and which stakeholders are regarded as the most important ones?
2. How are your customers/end-users represented in a development project? In which phases do the customer/end users provide input to the development?
3. To what extent do you experience that the customer actually understands what he needs in the early stages of a project?
4. What do you do if you see that there are alternative solutions which may serve the users better than the solutions which were originally specified?
5. Who are typical user representatives from the customer? Does the user representative understand the end-users' needs well enough? Why/why not?
6. During a project, how often are you in contact with the customer/end users of the product? Fixed intervals, occasionally or continuously throughout the project?

Documentation and criticality

1. To what extent do you document the development of software? Is the documentation process fixed or does it vary with the type of project you are involved in?
2. How does experience transfer occur throughout your organization? How is the experience transfer in specific between project teams or between individuals? How is learning shared throughout the organization and between projects?
3. Do you use too much, just right, or too little time on documentation. Please explain your opinion? When do you feel the documentation becomes a problem?
4. How is your organization adapted to development of software with high criticality? Do you need to change the development method if the end product might lead the loss of life/health?

Vulnerability and risk management

1. How much does individuals mean during development? How are the possibilities of finding substitutes in relation to sickness/absence/termination of employment?
2. Are personnel completely or only partly dedicated to specific projects in specific time periods? How is individuals' level of dedication determined?
3. Please define the personnel you consider to be key personnel in the software development process. How does the absence of key individuals affect projects? Is such absence considered to be a problem?

Personnel and culture

1. How is the typical project team put together regarding qualifications, experience, abilities? What kind of people? Unique people, visionary or "work horses"? Why are the teams put together this way?
2. How would you describe the people who are attracted to your organization? Are they seeking structure or chaos?

Do you think that there are important questions I failed to address? Please explain and elaborate...