NTNU
Innovation and Creativity

Use Cases in Practice:
A Study in the Norwegian Software Industry

Margrethe Adde Kjeøy
Gerd Melteig Stalheim

Master of Science in Computer Science
Submission date: June 2007
Supervisor: Tor Stålhane, IDI

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

Problem Description

Use Case is a technique for requirements management used in the software industry. In this master thesis we will investigate how the technique works in practice. We will interview developers, testers and project managers in Norwegian software companies about how they are using Use Cases and their opinions about the technique. Based on the results we will design a survey that will try to discover general attitudes to the Use Case technique. The survey will be sent to a selection of companies in the Norwegian software industry. Based on the response we will try to come up with improvement suggestions to the technique.

Assignment given: 19. January 2007
Supervisor: Tor Stålhane, IDI

Abstract

This Master's thesis investigates how project teams apply Use Cases and what problems they encounter with the employment of Use Cases by interviewing and surveying a number of Norwegian software companies. The thesis examines what developers and clients think is difficult and easy about Use Cases, how well the technique worked in a specific project, and how well the technique works in discussions with clients. A list of improvement suggestions for the Use Case technique is made based on the interviews, survey and literature study.

The key findings in this thesis are summarized as eight improvement suggestions. The three most important are: (1) that Use Cases should be supplied with user interface prototypes when used in discussions with clients, (2) that companies should make use of a tool that makes it easier to get the overview of related Use Cases, and (3) that one should avoid to write details about the user interface in Use Cases. Other findings are that Use Cases are most commonly used for requirements specification, estimation, programming and constructing test cases, and that it is difficult to find the right level of detail when writing Use Cases.

Preface

This report is a Master's thesis at the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU). The work with this thesis took place during Spring 2007.

We would like to thank our supervisor, Professor Tor Stålhane, for all help and advice this semester. We would also like to express our gratitude toward everyone that took their time to participate in our research and that gave us insight into how they work with Use Cases in their companies. We would also like to thank Terje Øfsdahl for taking his time to go through our survey and interview and give valuable feedback. Lastly, we thank Gyrd Norvoll for taking her time to go through our interview guide.

Trondheim, June 2007.

Margrethe Adde Kjeøy

Gerd Melteig Stalheim

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the motivation, objective and context of this thesis. The last section gives an overview of the structure of the thesis.

1.1 Motivation

Use Case is a requirements management technique in software development, and is used by many companies in the software industry. The technique was originally introduced to document the requirements in a way that all stakeholders could understand. Today, the technique is also used for a number of other purposes like programming, estimation and constructing test cases.

It is important that all stakeholders gain a common understanding of Use Cases and thus the system to be developed. In autumn 2006 we therefore investigated the comprehensibility of Use Cases. The research was conducted on students and the result was that students with no former experience with Use Cases had problems with the understanding. This result is useful and important, but since the research was conducted on students and not in a real software development project, it is important to investigate the results further in a real setting.

The motive for our research is therefore to investigate the application of Use Cases in a real setting, and since Use Cases are an important part of the software development process, our purpose is to find out how the Use Case technique can be improved based on this investigation.

1.2 Thesis Objectives

The objective for this thesis is to study how Use Cases are applied in the Norwegian software industry, and how well the technique works for different purposes. The purpose is also to investigate how the Use Case technique can be improved based on software developers experience with Use Cases.

We have formulated six research questions:

RQ1 When is it appropriate to apply Use Cases?

RQ2 For what purposes are Use Cases applied?

RQ3 How well did Use Cases work in a specific project?

RQ4 Do Use Cases work well in discussions with clients?

RQ5 What is difficult and what is simple about Use Cases?

RQ6 How can we improve the Use Case technique?

These research questions will be answered by interviewing and surveying a number of companies.

1.3 Thesis Context

This thesis has been carried out at the Department of Computer and Information Science at the Norwegian University of Science and Technology with supervision from Professor Tor Stålhane. The companies that participated in our research deliver consultancy services and own software products in the Norwegian software industry. All companies and persons that participated in our research are anonymous.

1.4 Thesis Outline

This report contains the following chapters:

Chapter 2 Software Engineering Processes
Describes development methodologies and the three processes of software requirements engineering.

Chapter 3 Use Cases in Software Requirements Engineering
Gives an introduction to Use Cases and Use Case formats, and describes two other requirements management techniques.

Chapter 4 Previous Research
Describes previous research conducted in the software industry and on students.

Chapter 5 Relevant Research Methods
Describes quantitative and qualitative research methods.

Chapter 6 Research Design
Presents the research goal and research questions for this thesis, the choice of research approach and how the interviews and survey in our research were designed.

Chapter 7 Operation of the Interviews and the Survey
Describes how the interviews and survey were conducted.

Chapter 8 Results from the Interviews
Presents the results from the interviews conducted in four companies.

Chapter 9 Results from the Survey
Presents the results from the survey in accordance with the research questions.

Chapter 10 Discussion of Results
Discusses the results and presents improvement suggestions.

Chapter 11 Discussion of Validity
Discusses the validity of our findings.

Chapter 12 Conclusion
Highlights the main conclusions of this thesis.

Chapter 13 Further Work
Presents how our work can be further elaborated.

Part I

Prestudy

Chapter 2

Software Engineering Processes

Software engineering is defined by The Institute of Electrical and Electronics Engineers (IEEE) as follows:

> The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software, (IEEE, 1990).

This chapter presents some important concepts in software engineering. Section 2.1 presents commonly used development methodologies and Section 2.2 describes three processes in requirements engineering.

2.1 Development Methodologies

A software development process is defined by IEEE as follows:

> The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use, (IEEE, 1990).

There exists several methodologies used in the software development process and this section gives a short description of the most important.

2.1.1 Agile Development Methodologies

Craig Larman describes agile development as follows:

> Agile development methods apply time-boxed iterative and evolutionary development, adaptive planning, promote evolutionary delivery, and include other values and practices that encourage *agility* - rapid and flexible response to change, (Larman, 2004).

This section describes three agile development methodologies: The Rational Unified Process (RUP), eXtreme Programming (XP) and Scrum.

Rational Unified Process

RUP is an iterative software development process framework, which is architecture centric and Use Case driven. It provides a disciplined approach to assigning tasks and responsibilities within a development organization and it is easy for project teams to adopt and justify the method to their needs. Figure 2.1, adopted from (Kruchten, 2003), shows the architecture of RUP.

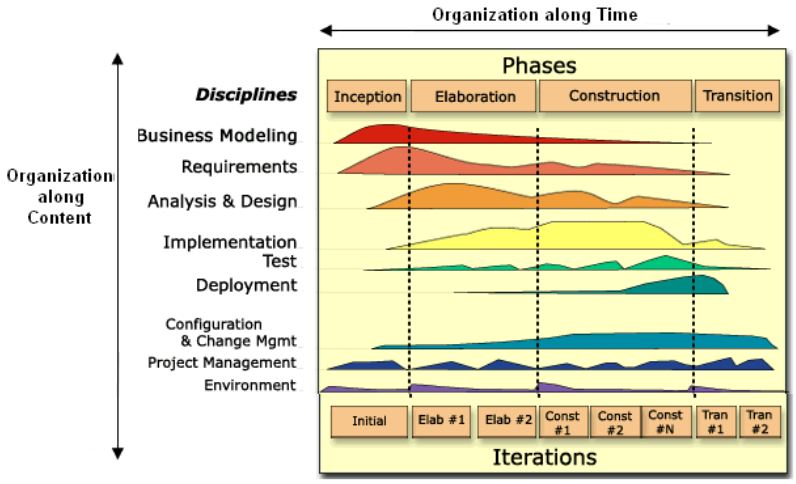

Figure 2.1: The architecture of RUP

The process has two dimensions; the time dimension and the organization along content. The time dimension shows the life cycle aspects of the process, while the organization along content groups the activities logically. As shown in Figure 2.1, each phase can have several iterations, and it shows when activities should start, and how much focus each activity should have in each phase. The RUP methodology focus on Use Cases and how they should evolve, be organized and how they form a basis for the development process. A more detailed description can be found in (Kruchten, 2003).

eXtreme Programming

XP is a well-known agile methodology developed by Kent Beck. The method is founded on four values: communication, simplicity, feedback, and courage (Beck, 1999). It is aimed at projects with duration of less than a year, and with small iterations of one to three weeks. User stories, further described in Section 3.10.1, are most commonly used in XP projects.

Figure 2.2 is adopted from (Wells, 2006) and shows the activities in an XP project.

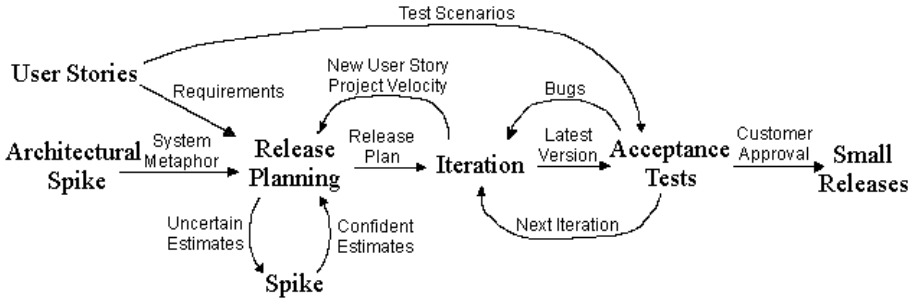

Figure 2.2: An eXtreme Programming Project diagram

Scrum

Scrum consists of three main phases called *Pregame* (planning and system architecture), *Game* (sprints) and *Postgame* (closure). A *sprint* is two to four weeks of development where a block of software is completed. The three phases are shown in Figure 2.3, which is adopted from (Schwaber, 1995).

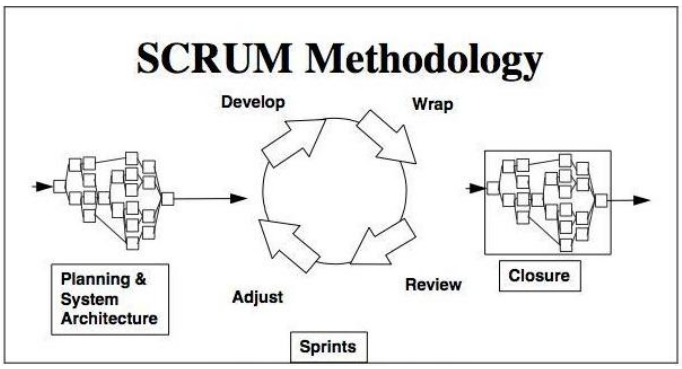

Figure 2.3: The SCRUM methodology

As with XP, Scrum uses User stories. Scrum does not prescribe a sequence in which the activities must be implemented, and a project can start with any activity, and can change between activities at any time. This increases the project's flexibility and productivity, (Schwaber, 1995).

2.1.2 The Waterfall Model

The Waterfall model is a sequential software development model, where the phases of requirements engineering, design, implementation, testing, and maintenance are performed in sequence. One should not proceed to the next phase until the current phase is completed. In each phase the obtained results are compared to the expected results of that phase, (Vliet, 2000). Figure 2.4 shows the concept of the waterfall model.

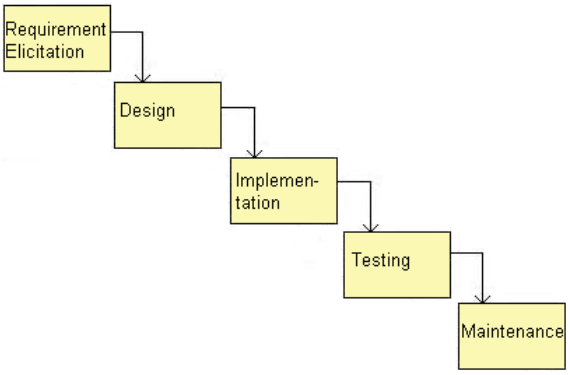

Figure 2.4: The Waterfall model

2.2 Software Requirements Engineering

Software engineering consists of a distinguishable number of phases: requirements engineering, design, implementation, testing and maintenance, (Vliet, 2000). This section describes the first part of the software engineering process, namely the software requirements engineering. Requirements engineering is one of the most critical steps in the software development process, and includes three processes; requirements elicitation, requirements specification and, requirements validation and verification, (Vliet, 2000).

2.2.1 Requirements Elicitation

Requirements elicitation is all about harvesting requirements from the stakeholders, and that is why the focus in requirements elicitation must be in gathering of information (Jenkins, 2005). The usual output is a requirements specification document which detail which of the requirements the project will address and, also, which it will not. Capturing the right requirements, and interpret them in the right way is crucial for the success of the project. Whether the project fails or not, depends on how well the project satisfies the client's needs.

> Success in software requirements, and hence success in software development, depends on getting the voice of the customer as close as possible to the ear of the developer, (Wiegers, 1999).

The most common techniques for requirements elicitation are Use Cases, interviews, questionnaires, and conversation, (Wiegers, 1999). Other techniques for requirements

elicitation are: introspection, protocol, interaction, and discourse analysis. It is proposed to use these techniques in combination, to take advantage of the strong points in all the methods, (Goguen and Linde, 1993).

2.2.2 Requirements Specification

As mentioned earlier, the output of the requirements elicitation process is a requirements specification document. This document can be of different types, according to which level of requirements that are elicited. There are three levels of software requirements, as well as several nonfunctional requirements. The highest level of requirements is the *business requirements* which represent high-level objectives of the organization or client requesting the system or product. These requirements are captured in a document that describes the vision of the project.

The next level of requirements is *user requirements* which describes tasks that the users must be able to accomplish with the product. These requirements are captured in Use Cases or scenarios.

The last level of requirements is the *functional requirements*, which define the software functionality the developers have to build, and are defined in the Software Requirements Specification (SRS). It is suggested to use Use Cases, SRS, or a combination of both, (Wiegers, 1999). SRS is further described in Section 3.10.2.

2.2.3 Requirements Validation and Verification

The last process in software requirements engineering consist of two phases: the *verification* and *validation*. In the verification phase, one checks if the system meets its requirements, while in the validation phase, one checks if the system meets the user's requirements. The requirements specification should reflect the mutual understanding of the problem to be solved, between the developer and client. To gain this understanding it is important to validate the requirements to see if everything has been described and if it has been described properly. The validation of the requirements thus means checking them for properties like correctness, completeness, ambiguity and consistency. It is important that the client participates in this validation process. The client is the owner of the problem, and the only one to decide if the requirements specification adequately describes their problem. Therefore it is important that the clients understand the contents of the requirements specification. The validation and verification techniques used are therefore often techniques that translates the requirements into a form that suits user inspection. Examples of such techniques are natural-language paraphrasing, prototyping and animation, (Vliet, 2000).

Chapter 3

Use Cases in Software Requirements Engineering

There exists several Use Case formats and this chapter gives an introduction to Use Cases and summaries the most important Use Case formats. Section 3.1 gives an introduction to Use Cases, Section 3.2 describes Use Case briefs, Section 3.3 describes Casual form, Section 3.4 describes fully dressed Use Cases, Section 3.5 describes the two-column table format and Section 3.6 describes UML Use Case diagrams. Section 3.7 describes the term Essential Use Cases, Section 3.8 explains why practitioners recommend to write Use Case descriptions instead of Use Case diagrams, Section 3.9 gives examples of how Use Cases can be supplied with other tools, and Section 3.10 presents two other requirements engineering techniques.

3.1 Introduction to Use Cases

The concept of Use Cases was first introduced by Ivar Jacobson in 1987 as a tool for modeling functional requirements, (Jacobson, 2003). The idea was quickly adopted worldwide after the publishing of the book *Object-Oriented Software Engineering: A Use Case Driven Approach*, (Jacobson et al., 1992), and has remained an important method for requirements management since. Today there exist a number of definitions of Use Cases, (Cockburn, 1997). Stated simply however, a Use Case describes how an actor interact with a computer system to achieve a goal.

Use Cases treat the system as a black box where all the interactions with the system are seen from the outside. They usually do not contain any technical jargon, but focus on using the language of the end user. The Use Case can take many forms, but as a minimum it contains a name and a basic course of action. A Use Case can be written either by the client, the developer or by the client and developer as a team. Use Cases connect many other requirements details and provide a scaffolding that connects information in different parts of the requirements. They are connected to other requirements as user interface requirements, user interface details and business rules, (Cockburn, 2001).

There exist two principal Use Case notations, i.e. as textual descriptions and as graphical representations. In this thesis we will call the textual notation *Use Case descriptions*, the graphical representation *Use Case diagrams*, and we will use the term *Use Cases* as a collective name for both.

A **Use Case description** is a textual description of a user interaction with a system. A typical Use Case is illustrated in Table 3.1 which is based on a template in (Cockburn, 2001).

«Name of the Use Case as a short active verb phrase name»

Primary actor: «Anyone or anything with behavior» **Scope:** «The system under discussion» **Stakeholders and interests:** «Someone or something with a vested interest in the behavior of the system under discussion» **Precondition:** «What must be true before the Use Case runs» **Minimal Guarantee:** «The fewest promises the system makes to the stakeholders» **Success Guarantee:** «States what interests of the stakeholders are satisfied» **Main Success Scenario:** «A numbered sequence of steps in which nothing goes wrong» **Extensions:** «What can happen differently during the scenario»

Table 3.1: A Use Case template

Most of the sections in Table 3.1 are self-explanatory, but some of them may need some extra explanation. *Scope* states the system under discussion that applies to this Use Case. The *Precondition* is a set of constraints that must be true before the Use Case starts. The *Main success scenario* usually consists of three to ten, numbered or bulleted, steps of user action. *Extensions* are alternative flows in the main success scenario. An extension starts with a condition, and continues with a sequence of action steps that happen under that condition.

A **Use Case diagram** is a graphical representation of a user's interaction with a system, as shown in Figure 3.1.

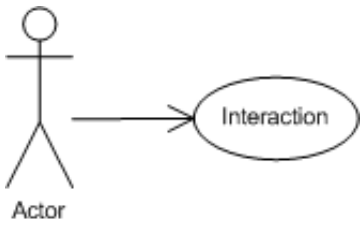

Figure 3.1: Use Case diagram

The stick-man represents an actor which may be either a person or another computer system interacting with the system. The text in the ellipse shows the interactions between the user and the system. Actors and ellipses are connected by specific relations.

As to the content and structure of Use Case descriptions and Use Case diagrams, the writer is free to use a style that fits his or her personal preferences and the situation at hand. There are many formats to choose from.

The most important advantage of Use Cases is that they describe a system in a manner that all stakeholders can understand. They are therefore used as a contract between stakeholders for the behavior of the computer system, (Cockburn, 2001). As a user-centered technique, Use Cases capture the requirements from the user's point of view, ensuring that the correct system is developed. Other benefits of using Use Cases are their usefulness in estimating, scheduling and validating effort, and that test cases can be directly derived from them. Use Cases contain a description of things that might go wrong, and projects benefit from having exceptions identified early because it saves time later in the project, (Firesmith, 1995).

A weakness of Use Cases is their lack of formality in the definitions of terms like *use case*, *actor*, *extends* and *includes*, (Firesmith, 1995), and since the Use Cases are written in narrative language, there are room for several interpretations of the text which may result in misunderstandings.

3.2 Use Case Brief

A Use Case brief is a short description of a Use Case behavior, and is useful for estimating work complexity. See Table 3.2 for a sample Use Case brief, adopted from (Cockburn, 2001).

Actor	**Goal**	**Brief**
Production staff	Modify the administrative area lattice	Production staff adds administrative area metadata (administrative hierarchy, currency, language code, street types, etc.) to the reference database. Contact information for source data is cataloged. This is a special case of updating reference data.

Table 3.2: Sample Use Case brief

For teams with good internal communication, Use Case briefs are enough for describing the behavioral requirements. For other teams, the Use Case briefs can be used as a starting point to the rest of the requirements document, (Cockburn, 2001).

3.3 Casual Form

An informal way of writing a Use Case is as a narrative, called **Casual form**. The Use Case is written in prose and describes at a high level how an actor interacts with the system to accomplish a goal. Table 3.3 illustrates a Use Case written in Casual form, (Cockburn, 2001).

Use Case name: Buy something
The requestor initiates a request and sends it to her or his Approver. The Approver checks that there is money in the budget, checks the price of the goods, completes the request for submission, and sends it to the Buyer. The Buyer checks the contents of storage, finding the best vendor for goods. The Authorizer validates Approvers signature. The Buyer completes request for ordering, initiates PO with Vendor. The Vendor delivers goods to Receiving, gets receipt for delivery (out of scope of system under design). The Receiver registers delivery, sends goods to Requestor. The Requestor marks request delivered. At any time prior to receiving goods, the Requestor can change or cancel the request. Canceling it removes it from any active processing (deletes it from system?). Reducing the price leaves it intact in processing. Raising the price sends it back to the Approver.

Table 3.3: Example of a Casual Use Case

The Casual form should be used early in the development or in situations where the development team are small and have close contact with their clients, (Cockburn, 2001).

Casual Use Cases are good for high-level summaries, but are not suitable for complex descriptions and can be ambiguous about who does what in the system, (Wirfs-Brock and McKean, 2001).

3.4 Fully Dressed Version

Table 3.4, adopted from (Cockburn, 2001), shows a fully dressed Use Case that is more formal than the Casual Use Case. The amount of information contained in the fully dressed Use Case varies depending on the situation, and every project team might define their own template. The most important thing is that the team provide one well-defined template that are used by everyone in the project, (Cockburn, 2001).

Name:	Buy something
Primary actor:	Requestor
Goal in context:	Requestor buys something through the system, gets it. Does not include paying for it
Scope:	Business - the overall purchasing mechanism, electronic and non electronic, as seen by the people in the company
Stakeholders and Interests:	Requestor: Wants what he/she ordered, easy way to do that. Company: Wants to control spending but allow purchases. Vendor: Wants to get paid for any goods delivered.
Precondition:	None
Minimal Guarantee:	Every order sent out has been approved by a valid authorizer. Order was tracked so that company can be billed only for valid goods received.
Success Guarantee:	Requestor has goods, correct budget ready to be debited.
Trigger:	Requestor decides to buy something.
Main Success Scenario:	1. *Requestor:* initiate a request. 2. *Approver:* check money in budget, check price of goods, complete request for submission 3. *Buyer:* check contents of storage, find best vendor for goods. 4. *Authorizer:* validate Approver's signature 5. *Buyer:* complete request for ordering, initiate PO with vendor 6. *Vendor:* deliver goods to Receiving, get Receipt for delivery (out of scope for design) 7. *Receiver:* register delivery: send goods to Requestor. 8. *Requestor:* mark request delivered.
Extensions:	1a. Requestor does not know vendor or price: Leave those parts blank and continue. 1b. At any time prior to receiving goods, Requestor can change or cancel request: Canceling it removes it from active processing (Delete from system?) Reducing rice leaves it intact in processing. Raising price sends it back to Approver. 2a. Approver does not know vendor or price: Leave blank and let Buyer fill in or callback 2b. Approver is not Requestor's manager: Still OK as long as Approver signs 2c. Approver declines: Send back to Requestor for change or deletion 3a. Buyer finds goods in storage: Send those up, reduce request by that amount and carry on.

CONTINUES ON THE NEXT PAGE

	3b. Buyer fills in Vendor and price, which were missing: Request gets resent to Approver 4a. Authorizer declines Approver: Send back to Requestor and remove from active processing. (What does this mean?) 5a. Request involves multiple Vendors: Buyer generates multiple POs. 5b. Buyer merges multiple requests: Same process, but mark PO with the requests being merged. 6a. Vendor does not deliver on time: System does alert of non-delivery. 7a. Partial delivery: Receiver marks partial delivery on PO and continues. 7b. Partial delivery of multiple-request PO: Receiver assigns quantities to request and continues. 8a. Goods are incorrect or improper quality: Requestor refuses delivered goods. (What does this mean?) 8b. Requestor has quit the company. Buyer checks with Requestor's manager: either reassign Requestor or return goods and cancel request.
Data Variations List:	None
Priority:	Various
Releases:	Several
Response Time:	Various
Frequency of Use:	3/day
Channel to Primary Actor	Internet browser, mail system or equivalent
Secondary Actors:	Vendor
Channels to Secondary Actors:	Fax, phone, car
Open Issues:	When is a canceled request deleted from the system?

Table 3.4: Example of a One-column table Use Case.

3.5 Two-column Table

Use Cases can also be written by dividing a table in two columns where the primary actor's actions are shown in the left column and the system's actions in the right column. This **two-column table** style was invented by Rebecca Wirfs-Brock, and came from the idea of a *conversation* between the actor and the system, see Table 3.5, (Wirfs-Brock, 1993). The two-column format clearly shows the system responses to the actions of the

Actor: Client	**System**
Enters order number.	
	Detects that the order number matches the winning number of the month.
	Registers the user and order number as this month's winner.
	Sends an e-mail to the sales manager.
	Congratulates the client and gives her instructions on how to collect the price.
Exits the system	

Table 3.5: Two-column Table Use Case

user, which is an advantage over the one-column format. Larry L. Constantine and Lucy A.D. Lockwood put it this way: "In this two-column format, the line down the middle represents, symbolically, the system boundary separating the user from the system. It is, in a sense the user interface. This format also highlights the part played by the user, which is the part most crucial to good user interface design". A drawback with this style is that it takes up a lot of space, (Constantine and Lockwood, 1999).

3.6 UML Use Case Diagrams

The Use Case formats described so far, are all versions of Use Case descriptions. Now we describe UML Use Case diagrams in detail. A more thorough description of UML Use Case diagrams can be found in (Fowler, 2003).

UML Use Case diagrams consist of actors and use cases (ellipses) which are connected by a link or a specific relation. The most common relations, in addition to the normal links, are «include», «extend» and generalization. Figure 3.2 illustrates a Use Case with normal links and the «include» and «extend» relation.

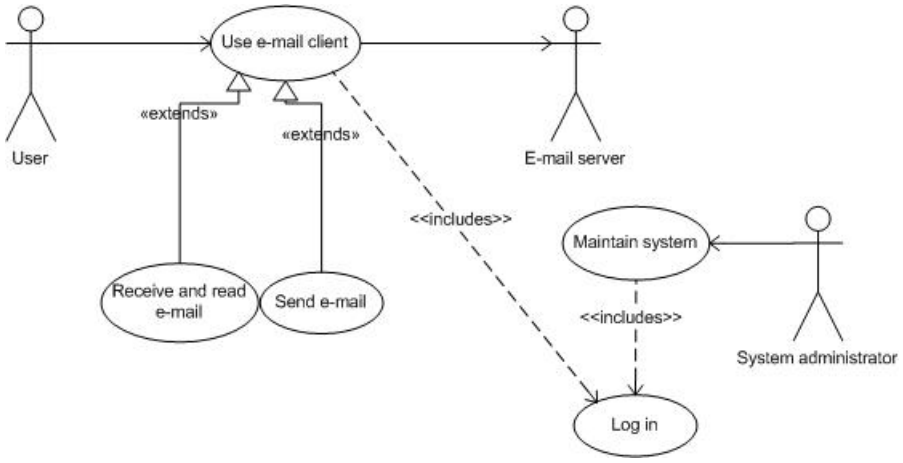

Figure 3.2: Example of a Use Case diagram with «include» and «extend»

The «include» relation is used when a behavior is similar across several use cases (ellipses) and we do not want to copy that description. The «extend» relation is used when describing a variation on normal behavior. Generalization is used when we have a use case that is similar to another use case but does a bit more, (Fowler, 2003).

The use of «include» and «extend» relations is a highly debated topic. There exist several opinions on how to use the relations, and whether they should be used at all. People seem to have problems with understanding «include» and «extend», and with separating the relations from each other because of similar notational symbol and meaning. It is therefore wise to reduce the number of «include» and «extend». It is recommended to concentrate on the «include» relation, and avoid the «extend» relation completely, (Cockburn, 2001).

To make a diagram more comprehensible, the higher level goals, namely the base use cases (ellipses), should be drawn higher than the included or extended use cases. This is intuitive to readers, and the arrow from a base use case to an included use case will always point down. To reduce the difficulty of separating the two relations, Cockburn suggests to use another arrow than the default UML drawing for «extend». The default is a dashed arrow (the same as «include») with the phrase «extend» along side it. By instead drawing an arrow that is completely different from the «include» arrow, one highlight the difference between the two, making it easier to understand, (Cockburn, 2001).

3.7 Essential Use Case Descriptions

This section do not focus on a Use Case format, but rather on the contents of Use Cases. Constantine and Lockwood, invented the term *Essential Use Case* while doing usability engineering. The primary idea behind Essential Use Cases is that assumptions about technology, such as the user interface, should be left out. The Use Case should instead focus on actor intent. "In particular, conventional use cases typically contain too many built-in assumptions, often hidden or implicit, about the form of the user interface that is yet to be designed", (Constantine and Lockwood, 1999).

Constantine and Lockwood define Essential Use Cases as follows:

> An Essential Use Case is a structured narrative, expressed in the language of the application domain and of users, comprising a simplified, generalized, abstract, technology-free and implementation-independent description of one task or interaction that is complete, meaningful, and well-defined from the point of view of users in some role or roles in relation to a system and that embodies the purpose or intentions underlying the interaction, (Constantine and Lockwood, 1999).

Constantine and Lockwood use the two-column table style. Therefore, the examples in this section appear in that format. Note however that Essential Use Case descriptions are not synonymous with the two-column table format. Any Use Case format can be written in an essential style. To understand the difference between Use Case descriptions and Essential Use Cases, consider the Use Case in Table 3.6.

Actor: Customer	**System**
Types name and password in dialog box. Presses the OK button.	
	Authenticates the user Displays a list of possible actions
Chooses an action by marking the wanted action with a mouse click. Presses the OK button.	

Table 3.6: Example of a Use Case description assuming user interface details

The Use Case description assumes details about the user interface. It assumes for instance that the user has to identify himself/herself by typing name and password and then pressing a button. It omits the possibility of having, for instance, a biometric authentication system. Now, consider the next Use Case description, written in an essential style, Table 3.7.

Actor: Customer	**System**
Identifies self	
	Authenticates the user Offer possible actions
Chooses an action	

Table 3.7: Example of an Essential Use Case description

The Essential Use Case description does not contain any assumptions about technology. It does not set any constraints on how the system should be developed, and the programmers can design the user interface without being restricted by the Use Case description. A major advantage of leaving user interface details out, is that the Use Cases do not have

to be updated every time the design changes. Another advantage is that the essential style allows brevity, as one can clearly see from the example above. Alistair Cockburn, Craig Larman and Donald Firesmith also recommend to write Use Case descriptions in an essential style, (Cockburn, 2001), (Larman, 1997) and (Firesmith, 1995).

3.8 Why Focus on Use Case Descriptions

Cockburn suggest focusing on *writing* Use Cases instead of drawing ellipses and stick-men. Cockburn substantiates on writing text by claiming that graphical notation suffer from two usability problems: that end users and business executives are not likely to be familiar with graphical diagrams and have little patience to learn, and that diagrams do not show all that one need to write, (Cockburn, 2001).

Larman agrees with Cockburn's viewpoint about writing text: "Use Case diagrams and Use Case relationships are secondary in Use Case work. Use Cases are text documents. Doing Use Case work means to write text", (Larman, 1997).

Cockburn and Larman's suggestion on focusing on text is in accordance with our own results from an experiment conducted on students in 2006. The result from the experiment showed that textual description is more comprehensible than graphical notation, (Stalheim and Kjeøy, 2006). A thorough description of the results from the experiment is given in Chapter 4.

3.9 Use Cases Supplied with Other Methods

To aid the Use Case work, some practitioners use remedies in addition to Use Cases. This section presents two examples of how Use Cases can be combined with other tools. The first example uses mockups, while the second uses role-play.

3.9.1 Use Case Workbench

Use Case Workbench is a tool for Use Case engineering. Figure 3.3, adopted from (Nawrocki and Olek, 2005), illustrates a mockup generated by Use Case Workbench. The mockup is supposed to animate the Use Case description, and present functionality by combining Use Case descriptions with screen design.

The mockup is based on a web browser and consists of two frames; the scenario window and the screen window. The left side of Figure 3.3 shows the Use Case description where the bold step represents the step which belongs to the screen design shown on the right side of the figure. By using this mockup, the stakeholders get the opportunity to visualize ideas of how the interface will look like during the steps in the Use Case description. This will make it easier to elicit requirements and gaining agreement on the system, which in turn will make the client-developer communication more efficient, (Nawrocki and Olek, 2005).

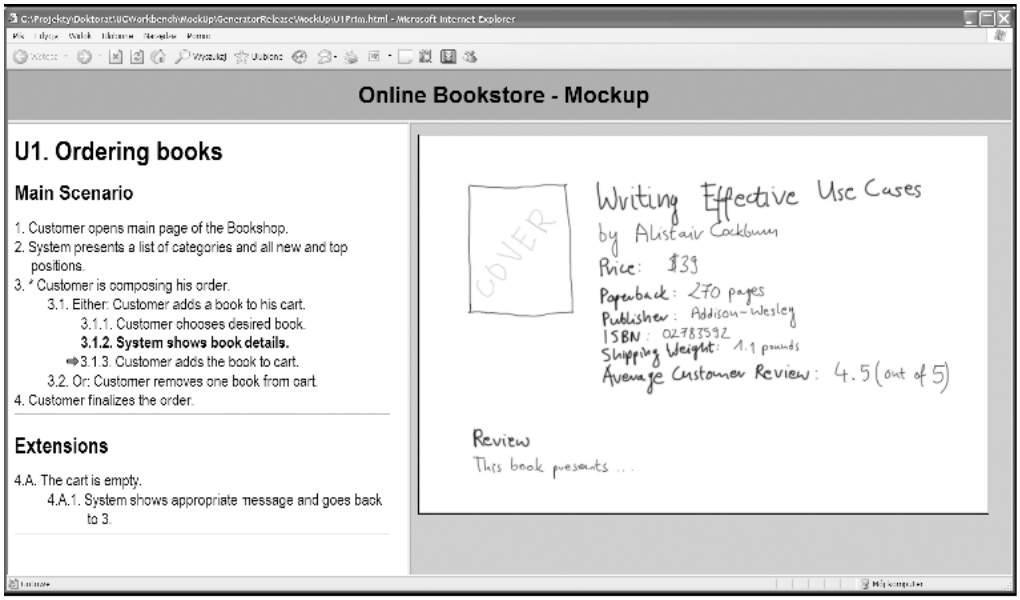

Figure 3.3: An example of a mockup screen

3.9.2 Role-play and Use Case Cards

To make the Use Case development more accessible and better guided, Robert Biddle, James Noble and Ewan Tempero present a technique that uses role-play and index cards, (Biddle et al., 2001). Figure 3.4 illustrates a team performing a roleplay. The technique is based on the established CRC card technique that document collaborative design decisions on index cards, (Beck and Cunningham, 1989).

Figure 3.4: A team performing roleplay

The Use Case descriptions are presented in a two-column format style as in Table 3.7. The starting point is to give names to the Use Case descriptions and when all the Use Case descriptions have been identified, the next step is to iteratively work out the body of the Use Case description. This is done with a role-play that involves two people where one act as the system and the other as the user. For a more thorough description of how this technique works, see (Biddle et al., 2001).

3.10 Other Requirements Engineering Techniques

This section gives an overview of two other techniques used in requirements engineering. These techniques are User stories and SRS.

3.10.1 User Stories

A User story describes functionality that is valuable to a user of a system. User stories were first introduced in eXtreme Programming (XP) as a way of expressing requirements. The story is hand-written on a card. This card is the visible manifestation of the User story, but the conversation where the details are worked out is the most important, (Cohn, 2004b).

There are several reasons to why User stories should be used. Among the arguments is that User stories emphasize verbal communication, they are comprehensible to everyone, and they encourage participatory design. This gives a better communication between the developing team and the client, and avoids different interpretations among the participants, (Cohn, 2004b).

The User story technique is effective in small project teams where the client is near the developers, but lack formality if used in development of high-critical systems. In these situations, Use Cases are more appropriate (Cockburn, 2001).

The main differences between Use Cases and User stories are the scope, level of detail and purpose. Use Cases have a main success scenario, which refers to the primary successful path through the Use Case. Each path through the Use Case is referred to as a scenario. The scope of the User stories is different. One User story is similar to a single scenario of a Use Case.

The purposes for which Use Cases and User stories are written, are also somewhat different. While the Use Cases have the purpose of documenting an agreement between the client and development team, the stories are written to facilitate release and iteration planning, and serve as placeholders for conversations about the user's detailed needs. Use Cases are written as the result of an analysis activity, while User stories are written as notes that can be used to initiate analysis conversations, (Cohn, 2004a).

3.10.2 Software Requirements Specification

IEEE has developed a standard for how to write a good Software Requirements Specification (SRS), which can be found in (IEEE, 1994). In this standard the system is described as a set of "The system shall ...."-sentences that focus on what functions the system shall support, in contrast to Use Cases that describes how the system is used by a user. Writing requirements compliant to this standard often result in tedious and boring reading. This might result in the specification not being read carefully enough, and the project team not getting enough information about the requirements from the client, (Cohn, 2004a). The IEEE framework for the requirements specification is especially appropriate in classic

models of the software development process; the waterfall model and its variants, (Vliet, 2000).

The SRS constitutes the agreement between developers and clients on the content of the product that is going to be built. It should be structured and written in a way that is easy to read and understand. "As the ultimate repository for the product requirements, the SRS must be comprehensive: *all* requirements should be included". Neither developers nor clients shall make assumptions, all the functionality shall be accurately described and agreed upon, (Wiegers, 1999).

Chapter 4

Previous Research

Previous research has investigated the comprehensibility and application of Use Cases. Some research has been conducted in a real software development setting while other research has been conducted on students. This Chapter describes the findings in both types of research. Section 4.1 describes previous research in the software industry, and Section 4.2 describes research conducted on students.

4.1 Previous Research in the Software Industry

This section describes the findings of a survey that identified how Use Cases are employed in practice are described, then ten problems about using Use Case diagrams in real projects are described.

4.1.1 A Survey of Use Cases in Practice

In *Use Cases in Practice: A survey*, the authors attempt to find out how Use Cases are employed by developers. The most important result from this survey was that industrial practices place emphasis on the coupling between Use Cases and user interface details even though this is not recommended. The authors suggest to use task models as a complementary to Use Cases. A task model specifies what the user does, or wants to do, and why, and is similar to Use Cases. In contrast to Use Cases the tasks in task models are decomposable into subtask and atomic actions. Based on the task model, the user interface may be automatically generated. Another issue was that the participants in this survey had problems modeling and understanding the «include», «extend» and generalization relationships, (Sinnig et al., 2005).

4.1.2 Top ten Problems from Real Projects Using Use Case Diagrams

In *How to Avoid Use-Case Pitfalls*, Susan Lilly focuses on problems with Use Case diagrams based on observations from a number of real projects. Table 4.1 lists the problems, (Lilly, 2001).

Top ten problems from real projects

Problem 1:	The system boundary is undefined or inconsistent.
Cure:	Be explicit about the scope, and label the system boundary accordingly.
Problem 2:	The Use Cases are written from the system's (not the actor's) point of view.
Cure:	Name the Use Cases from the perspective of the Actor's goals.
Problem 3:	The actor names are inconsistent.
Cure:	Get agreement early in the project and establish a glossary to define the actors.
Problem 4:	Too many Use Cases.
Cure:	Make sure the granularity of the Use Cases is appropriate.
Problem 5:	The actor-to-use-case relationships resemble a spider's web.
Cure:	The actors should not be defined too broadly.
Example:	Employee is general, while Phone clerk is more specific.
Problem 6:	The Use Case specifications are too long.
Cure:	The granularity of the Use Cases may be too coarse.
Example:	"Use schedule" is too broad, while "View schedule" and "Create schedule" are shorter and easier to understand.
Problem 7:	The Use Case specifications are confusing.
Cure:	Include a context field in your Use Case specification template to describe the set of circumstances in which the Use Case is relevant.
Problem 8:	The Use Case does not correctly describe functional entitlement.
Cure:	Make sure that each actor associated with a Use Case is completely entitled to perform it. If an actor is only functionally entitled to part of the Use Case, the Use Case should be split.
Problem 9:	The client does not understand the Use Cases.
Cure:	Teach them just enough to understand. Put a short explanation in the document, lead a training course and think long about using «include» and extend.
Problem 10:	The Use Cases are never finished.
Cure:	Loosely couple user interface details and Use Case interactions.

Table 4.1: Problems from real projects

The author says the reason for problem seven in the table: *The Use Case specifications are confusing*, is that the Use Case diagrams lack context. This can be fixed by including a Context field in the Use Case diagram where you describe the circumstances where this Use Case diagram is relevant. Another problem is that the steps in the normal flow look like a computer program. You avoid this by moving out conditional behavior to alternative sequences, and by avoiding to describe algorithms in the Use Case diagram and moving them to another document, (Lilly, 2001).

The author describes five symptoms that make the Use Case diagrams less understandable by the client (problem nine). The first symptom is that the client do not know anything about Use Case diagrams. The author suggest to fix this problem by explaining the Use Case diagrams in a short document, having a training session with the client, and avoiding «extend» and «include» as much as possible.

Another symptom mentioned by the author is that the Use Case diagrams do not tell a story. The fix to this problem is to contain a Context section in the Use Case template. The third symptom is that the organization of the Use Case diagrams does not match the way the client thinks of the system. To fix this problem, it is suggested to listen to how the client describes the business. The fourth symptom mentioned is that the Use Case diagram is written with technical words that are not part of the client's vocabulary. This must be avoided.

The first symptom described by the author to problem ten: *The Use Cases are never finished*, is that the Use Case diagrams have to change every time the user interface changes. This can be avoided by coupling the Use Case diagrams *loosely* to the user interface, not overdo it because the user interface design is likely to change as time goes by. The second symptom is that the Use Case diagrams changes every time the design of the system changes. The fix to this is to not contain design details in the Use Case diagrams. The third symptom is "analysis paralysis" of the alternative steps. This can be avoided by stop looking for more alternative sequences at some time, try to cover just 80% of the cases. For a more detailed explanation of all ten problems, see (Lilly, 2001).

4.2 Empirical Research Conducted on Students

Several researchers have conducted experiments on students to study the Use Case technique. This section summaries the key findings of relevant experiments. First other researcher's work is described, then our own research from fall 2006 is described.

4.2.1 Other Researcher's Empirical Research

Anna Bobkowska has reported some problems regarding Use Case diagrams. Among these were the stick-man notation that are not intuitive to use for representing computers and that the direction of the «extend» and «include» arrows are confusing, (Bobkowska, 2005).

The result from Karl Cox' experimentation is also that people have problems with understanding the differences between «include» and «extend» relationships, and how to use them, (Cox, 2000). In *Quality and Understandability of Use Case Models*, the authors perform an experiment on students with the aim to detect effects of guidelines when writing Use Cases. The result from this experiment indicated that guidelines based on templates constructs Use Cases that are easier to understand than guidelines without specific details on how to document each Use Case, (Anda et al., 2001).

4.2.2 Results from Depth Study 2006

In fall 2006 we performed an experiment on students that focused on the readers intelligibility of both Use Case diagrams and Use Case descriptions. The subjects were divided into two groups, where the subjects in one group had no experience with Use Cases and the subjects in the other group had learned about Use Cases in a course. The aim of the study was to find out whether there are differences in the understandability of Use Case diagrams and Use Case descriptions, and also whether there are differences in how well a person that have learned about Use Cases and a person that has never heard about Use Cases before understand Use Case descriptions and Use Case diagrams.

According to the results we have reasons to believe that Use Case descriptions are easier to understand than Use Case diagrams. The main reason for this is that it seems to be hard to understand the meaning of the «include» and «extend» relationships in Use Case diagrams. Statements like, "I thought the arrow was supposed to point the other way" regarding the notational symbol of «include», supports findings that have been reported about Use Case diagrams earlier.

There were some differences between experienced and inexperienced readers when it comes to Use Case descriptions, because some inexperienced readers had problems with understanding *extensions* in Use Case descriptions. According to these findings, it is reasonable to claim that the *extensions* in Use Case descriptions, and the «include» and «extend» in Use Case diagrams, should be used with care, (Stalheim and Kjeøy, 2006).

Chapter 5

Relevant Research Methods

This chapter defines relevant concepts of this thesis. Section 5.1 gives a short introduction to research in software engineering, Section 5.2 and Section 5.3 explains quantitative and qualitative research approaches supplied with examples, respectively.

5.1 Research and Software Engineering

In order to perform research in software engineering, it is useful to understand the methods that are available. The methods in the software engineering field are: *The scientific method* where the world is observed and a model is built based on the observation, *the engineering method* where the current solutions are studied and changes are proposed, and then evaluated, *the empirical method* where a model is proposed and evaluated through empirical studies, for example, case studies or experiments and *the analytical method* where a formal theory is proposed and then compared with empirical observations. In this thesis we make use of the empirical method. There exist three empirical strategies which are commonly used in software engineering research, namely experiments, case studies and surveys. Empirical research can be of a qualitative or a quantitative nature, (Wohlin et al., 2000).

5.2 Quantitative Research Methods

Quantitative research develop metrics (numbers) that can describe a phenomena under study. The data is then analyzed by statistical methods. The aim of quantitative research is to identify a cause-effect relationship while qualitative research are appropriate to find out why the results from a quantitative investigation are as they are. Therefore, the two approaches should be regarded as complementary rather than competitive, (Wohlin et al., 2000). This section describes two approaches to quantitative research, namely experiments and surveys.

5.2.1 Experiments

An experiment is a research activity that is undertaken within controlled conditions. Subjects are assigned different treatments, and one or more variables are manipulated while the others are controlled. Then the effect of the treatments are measured and analyzed with statistical methods.

Experiments may be performed off-line in a laboratory, or on-line where the investigation is executed in the field under normal conditions. In both types of experiments, we identify the variables and sample over them which means that we select objects representing a variety of characteristics that is typical for the organization and design the research so that more than one value will be measured for each characteristic. The strengths of experiments are that we have high execution and measurement control, and that experiments are easy to replicate. A weakness is that the cost is high, (Wohlin et al., 2000).

5.2.2 Surveys

Surveys can be of both quantitative and qualitative nature and can take form as an interview or a questionnaire. Surveys are further described in Section 5.3.2.

5.2.3 Quantitative Measurements

Measurements are crucial for performing experiments. The measurements map the attribute of an entity to a measurement value. The mapping may be made in different scales. The most common scale types are nominal scale (names or symbols), ordinal scale (ranking after an ordering criterion), interval scale (same as ordinal scale but there is a notion of relative distance) and ratio scale (used when there exists a zero value and the ratio between two measures is meaningful). The interval and ratio scales are related to quantitative research. Measures can also be divided into objective measures where there is no judgment in the measurement value, and subjective measure where a person make a judgment of the measure, (Wohlin et al., 2000).

5.3 Qualitative Research Methods

The difference between quantitative and qualitative methods is that qualitative research avoid metrication and instead use other means of analyzing data. It is usually based on words rather than numbers, (Cornford and Smithson, 2006). This section describes four approaches to qualitative research, namely affinity diagrams, surveys, case studies and observation.

5.3.1 Affinity Diagram

An affinity diagram, also called "KJ" (named after Kawakita Jiro), is a method that gathers ideas and opinions about a topic and organizes them into groups. The diagram is made by having a group of people generating the ideas and writing each idea on a piece of paper. The ideas are then placed on a white-board for everyone to see. Then the participant

are free to move the ideas and place them in appropriate groups. The technique is especially useful when it is essential with innovative thinking and of importance to achieve consensus, (Wedde, 2000).

5.3.2 Surveys

Surveys as a data collection method has the advantage that it allows the researcher to obtain views or data from a large number of organizations or individuals in a limited time period. One disadvantage of using surveys is the response rate. A response rate of 20 per cent is seen as a good response, (Cornford and Smithson, 2006).

A survey can be both qualitative and qualitative depending on how it is designed. It can take form as an interview or a questionnaire. A survey is often an investigation of a tool or technique that has already taken place or before it is introduced. The investigation is performed on a sample that is representative for the population under study. The motive for surveys can be one of the following: descriptive, explanatory or explorative. Descriptive surveys describes a situation or some characteristics of a population. An explanatory survey makes claims about the population, while explorative surveys are used as a prestudy to a more thorough investigation, (Wohlin et al., 2000).

Interviews

The advantages of performing the survey as an interview is that it is possible for the interviewer to observe and ask further questions. A disadvantage is that it costs more and takes more time, (Wohlin et al., 2000).

It is important to plan interviews in advance, and to be aware of how to behave in a interview setting to get most out of the interview. A good interviewer is ,among other factors, structured, precise, friendly and takes control of the interview, (Kvale, 1996).

Questionnaires

Questionnaires could be in electronic form or on paper. Advantages are that it is cheaper and takes less time. The disadvantages are that we get more "do not know" answers, and that the response rate is lower than with interviews, (Wohlin et al., 2000).

When making questionnaires it is important to consider the question wording and design of the questionnaire, (L.Mordal, 1989).

5.3.3 Case Study

Case studies investigate a single phenomenon within a specific time space. It can be used to evaluate how or why certain phenomena occur or to evaluate the differences between two methods. The difference from experiments is that case studies study the specific situation. A disadvantage with case studies is that the data are harder to interpret and generalize, (Wohlin et al., 2000).

Another limitation is the lack of control of individual variables and the difficulty of locating causality. The main strength of the method is the richness of data that are obtained when restricted to a single situation, (Cornford and Smithson, 2006).

5.3.4 Observation

Observation is to observe subjects in their normal work situations to get insight into how a tool is used or how a work process is, (Cornford and Smithson, 2006).

Part II

Research Planning

Chapter 6

Research Design

This chapter presents the design of our research. Section 6.1 presents the research goal and research questions for this thesis, Section 6.2 explains the choice of research method, Section 6.3 describes how the subjects for this study were selected, Section 6.4 describes the design of the interviews with developers and clients, and Section 6.5 describes the design of the survey.

6.1 Research Goal and Research Questions

Table 6.1 presents the research goal and research questions of this thesis.

Research goal:	
Investigate how Use Cases are applied in the Norwegian software industry and how well the technique works for different purposes. In addition, investigate how the Use Case technique can be improved based on developers and clients experience with Use Cases.	
Research questions:	
RQ1	When is it appropriate to apply Use Cases?
RQ2	For what purposes are Use Cases applied?
RQ3	How well did Use Cases work in a specific project?
RQ4	Do Use Cases work well in discussions with clients?
RQ5	What is difficult and what is simple about Use Cases?
RQ6	How can we improve the Use Case technique?

Table 6.1: Research goal and questions

RQ1 When Is it Appropriate to Apply Use Cases?

Use Cases are, as any other method, appropriate in some settings and not appropriate in others. Therefore, we wanted to find out when companies apply Use Cases and when they do not apply Use Cases, and the reasons for their choices. Based on the answers, it was of interest to come up with advices for when it is appropriate to apply Use Cases and when it is not. In addition, we wanted to find out what other requirement management techniques are in use, and which method is the one most preferred by developers. This is important to investigate because it is useful for project teams to have directions for when to apply Use Cases and when not to apply Use Cases.

RQ2 For what Purposes Are Use Cases Applied?

Use Cases can be applied in many ways and for many purposes, for example testing, estimating, system documentation and so on. It was of interest to investigate for what purposes companies apply them to see all possible ways Use Cases can be used. This is interesting because companies can be inspired by the list of applications and start to apply Use Cases in new ways, and can therefore make most of the Use Case technique.

RQ3 How Well Did Use Cases Work in a Specific Project?

It was of interest to investigate the use of Use Cases in a specific project and get opinions from both clients and developers about Use Cases. This was interesting to investigate because it would give us rich insight into how the Use Cases worked in a real project.

RQ4 Do Use Cases Work Well in Discussions with Clients?

Use Cases were first introduced as a tool that should aid the communication between developers and clients, but experiments have shown that Use Cases are not as easy to understand for non technical persons as first expected (see Chapter 4). It was of interest to investigate this further in a real setting. This is important to find out because if there are differences in the understanding, this means that the stakeholders do not have the same understanding of the system to be developed, which in turn may result in large costs later in the project.

RQ5 What Is Difficult and what Is Simple about Use Cases?

It was of interest to investigate what developers and client representatives think is difficult and simple about Use Cases on a general basis. It was of interest to see if there are any differences between developers and client representatives, but it was also of interest to see if there were differences between testers, programmers and project leaders. It is important to investigate what is difficult and simple about Use Cases because from the results it would be possible to set directions so that they become easier to apply.

RQ6 How Can We Improve the Use Case Technique?

It was of interest to gather experience with Use Case from people in the software industry and use this as a basis to come up with improvement suggestions to the Use Case tech-

nique. This is important because it is common to face problems with the application of Use Cases and it is therefore useful with suggestions for how to improve the technique.

6.2 Choice of Research Method

Based on the research questions stated in Section 6.1 we wanted to do an explorative study to get deep insight into how Use Cases are used in a small selection of companies. We thus needed to use a data collection method that gave as much information as possible from few people. Based on that criterion we chose to do interviews. Section 5.3.2 discusses some advantages and disadvantages when using this method.

After the interviews we did a descriptive study to determine the distribution of how Use Cases are used. To do this we generated a questionnaire based on the information gathered from the interviews. We used the information we got from the interviews to make questions and alternatives for the questionnaire. The reason we made a questionnaire in addition to the interviews was that we could reach more companies with the questionnaire than with the interviews. A discussion of advantages and disadvantages of questionnaires can be found in Section 5.3.2. From now on we will refer to our questionnaire as survey.

6.3 Selection of Subjects

The selection of subjects was based on a convenience sampling, (Wohlin et al., 2000). From our own personal contacts and those of our teaching supervisor we got hold of interview subjects in four companies. From a student organization we got hold of subjects to our survey in twenty two companies in Norway. We also interviewed two clients that we got hold of through one of the companies we interviewed.

6.4 Design of the Interviews

This section describes the design of the interviews with developers and the interviews with the client representatives.

6.4.1 Design of the Developer Interviews

This section describes the design of the interviews with developers. The interview was divided into six parts as described in this section. The relation between the six parts of the interview and the research questions are also described here. Appendix A contains the whole interview guide in Norwegian.

Part I: Personal Information

This part was not directly related to any of the research questions, but was necessary so the rest of the interview could be put into a context. The first four questions focused on basic information of the interviewee: The name of the interviewee, the number of years

spent in the industry, the interviewee's position in the company and the interviewee's responsibilities in the company. The last two question dealt with how many projects the interviewee had been working on that used Use Cases, and how he or she was involved with Use Cases, i.e if he or she had been taking part in the writing of Use Cases or just read them.

Part II: Information about the Company

As for part one, part two was not directly related to any of the research questions. Part two gathered basic information about the company: What kind of company the interviewee worked in, and what kind of clients the company had.

Part III: Application of Use Cases in the Company

This part was related to *RQ1 When is it appropriate to apply Use Cases?*, and *RQ2 For what purposes are Use Cases applied?*. It focused on how Use Cases were applied in the company: whether the company had guidelines for how to write Use Cases, and what kind of guidelines this was, and for what objectives the company applied Use Cases. It also dealt with general reasons for why the company applied Use Cases in some settings and why not in other settings.

Part IV: Application of Use Cases in one Particular Project

This part was related to *RQ3 How well did Use Cases work in a specific project?*. It focused on how Use Cases were applied in one particular project. First it focused on basic information for that particular project: what type of system was developed, what role the interviewee had in the project, what development methodology they used, how the requirements gathering took place, how many persons wrote and used the Use Cases and what responsibilities these persons had in the project.

Then information about what objectives they applied Use Cases in the project was collected, i.e for testing purposes, to help communication or so on, and how the interviewee felt that this application of Use Cases worked out. One question also focused on the interviewee's opinions of Use Cases as a communication tool. Then the focus was on how the Use Cases were written, how modifications in the Use Cases were handled in the project and reasons for why Use Cases were used.

Part V: Personal Opinions of Use Cases

This part was related to *RQ1 When is it appropriate to apply Use Cases?*, *RQ4 Do Use Cases work well in discussions with clients?* and *RQ5 What is difficult and what is simple about using Use Cases?*. It focused on the interviewee's personal opinions about Use Cases. It dealt with opinions about how interviewees thought Use Cases should be applied, what they thought was *difficult* about writing, reading and generally using Use Cases, and what they thought was *easy* about writing, reading and generally using Use Cases.

It also dealt with what requirements management techniques were preferred, and the reasons for the preferences. It also investigated what types of projects interviewee's thought that Use Cases should be applied in, and in what types of project they should not be applied in.

Part VI: Personal Experience with Use Cases

This part was related to *RQ6 How can we improve the Use Case technique?*. It investigated the interviewee's personal experience with Use Cases. It dealt with positive and negative experiences with Use Cases, and looked for ways to remove the negative efforts. It also dealt with supporting materials to Use Cases.

6.4.2 Design of the Client Interviews

This section describes the two parts of the interviews with two testers from a client company. The interview was related to *RQ3 How well did Use Cases work in a specific project?*, *RQ4 Do Use Cases work well in discussions with clients?* and *RQ5 What is difficult and what is simple about Use Cases?*. Appendix A contains the interview guide in Norwegian.

Part I: Personal Information

Part one of the interview focused on personal information of the interviewee: the name of the interviewee, the interviewee's position in the project and the employment in the company. It also dealt with their former knowledge of Use Case, and if they had participated in a similar project before.

Part II: Application of Use Cases

Part two focused on how Use Cases were applied by the interviewee and what the interviewee thought of the application. It also dealt with how helpful the Use Cases were, what was difficult and what was simple with the Use Cases, and what the interviewee thought of the length, layout, level of detail and numbering of the Use Cases. It also focused on if it would have been helpful with some kind of supporting materials in addition to the Use Cases.

6.5 Design of the Survey

This section describes the six parts of the survey and how they are related to the research questions. Appendix B contains the survey in Norwegian.

Part I: Personal Information and Information about the Company

The first part focused on personal information and information about the company. The respondents were asked to fill in their name, although this was optional. They were also asked to fill in the name of the company, their position in the company, and the number of years spent in the industry.

The five last questions were: responsibility in the company (testing, programming, requirements specification, project managing), number of projects they had participated in that applied Use Cases, how they had been involved with Use Cases (written them, written test cases from them, tested directly from them, programmed from them), type of Use Cases applied in the company (Use Case diagrams, Use Case descriptions, or both), and what type of development methodology they used in the company.

Part II: Application of Use Cases in the Company

This part was related to *RQ1 When is it appropriate to apply Use Cases?* and *RQ2 For what purposes are Use Cases applied?*.

The first question was an open question about what other responsibilities the person(s) who write Use Cases have. The next five questions focused on what tools the company use to write Use Cases in, and how they relate Use Cases to other documents. This was important because it was interesting to know if they used a specialized tool or just a normal text editor. Later in the survey the respondents were asked whether they would have preferred a specialized tool for Use Cases.

It was also of interest to know what processes (estimation of the project, to produce test cases, or so on) the Use Cases aided in the company. There were six alternatives to this question and the respondents could check as many as they wanted. They also had the opportunity to write their own answers. The respondents were also asked to check what type of aids they use in addition to Use Cases. To this question we came up with four alternatives based on the interviews. They were also allowed to fill in their own answer if none of the alternatives fitted. Then they were asked to check for either client representatives, developers or testers to the question on what persons Use Cases were primarily written for. If the respondents had written Use Cases him or herself, he or she was asked to describe how they proceeded when writing them.

The respondents were then asked to describe what parts their Use Cases usually contain. Based on templates we set up alternatives like *actor*, *main flow* and *pre-conditions*. The respondents were asked to check the alternatives that their Use Cases contain. To get a deeper understanding we asked them whether the Use Cases contained information about the user interface. The last question in this part was about how many pages an average Use Case is.

Part III: Use Cases as a Tool for Communication with the Client

This part was related to *RQ4 Do Use Cases work well in discussions with clients?*. On a scale from one, *I strongly disagree*, to five, *I strongly agree*, the respondents were asked to consider seven assertions about how appropriate Use Cases are as a tool for communication with client representatives. The assertions are listed below:

1. *Use Cases work well as a tool for communication with client representatives.*
2. *I think the Use Case technique works well as a way to communicate requirements when the client representative is not so familiar with technology.*
3. *In addition to Use Cases one should use user interface prototypes because client representatives needs a picture of what the system will look like.*
4. *The ideal solution is to apply Use Cases between developers, and user interface prototypes for client representatives.*
5. *I have received feedback from client representatives that they would rather talk about user interfaces than Use Cases.*
6. *I have received feedback from client representatives that they think it is a waste of time to apply Use Cases.*
7. *I have received feedback from client representatives that they think it is difficult to get the overview of all Use Cases.*

Part IV: Personal Opinions about Use Cases

This part was related to *RQ1 When is it appropriate to apply Use Cases?* and *RQ5 What is difficult and what is simple about Use Cases?* The respondents were asked to consider eight assertions about things that are difficult about Use Cases on a scale from one, *I strongly disagree*, to five, *I strongly agree*. For example *I think it is difficult to find the right level of detail that suits developers, testers and client* and *I think it is difficult to keep the Use Cases updated throughout the project*. The rest of the assertions can be found in question 4.2 in Appendix B.

They were then asked to consider on the same scale if they thought that the time spent on Use Cases (writing, reading and updating) was appropriate, and that the level of detail was appropriate, and similarly if the time spent keeping the Use Cases updated was appropriate.

Then they were asked to consider on the same scale in what types of projects Use Cases should be used: development of completely new systems, further development of old systems, Internet portal projects, and projects where one works near the client representatives. Next, the respondents had to consider seven general assertions about Use Cases, for example *I like working with Use Cases*, *To work with Use Cases is boring*, *I prefer other methods than Use Cases* and so on, on the same scale.

The last section of this part focused on how many times the respondent had experienced problems with Use Cases: *I have experienced that we have spent so much time on the Use Cases in the beginning of a project that when the development started, the Use Cases have become irrelevant*, *I have experienced that Use Cases have become so extensive that I did not bother to read through the whole Use Case*, and so on. The alternatives to these questions were *Never experienced*, *Experienced once* and *Experienced two times or more*.

It was also interesting to know if the respondent preferred other methods than Use Cases, and in that case: which method. They were asked to give reasons for their answers to this question. The respondents were also asked what Use Cases should *not* be used for. They were also asked to give an answer to which person(s) in a project should write Use Cases, and the alternatives were *A Use Case specialist*, *Developers* or *The client*. It was allowed to check more than one alternative in case they mean that it should be written by a group of persons.

The last question of this part was about what the respondents thought Use Cases are helpful for. The alternatives were *To estimate a project*, *To structure a project*, *To get good requirements*, *To get an introduction to a system I do not know in advance*, *To make a training course*, *To write user documentation* and *To communicate with the client*. It was allowed to check more than one alternative.

Part V: Personal Experience with Use Cases

Part five was related to *RQ1 When is it appropriate to apply Use Cases?* and *RQ5 What is difficult and what is simple about Use Cases*. It contained two open questions. The first was: *Please describe some positive experience with Use Cases*, and the second was: *Please describe some negative experience with Use Cases*.

Part VI: Improvement Suggestions to the Use Case Technique

Part six was related to *RQ6 How can we improve the Use Case technique?*. On a list of five improvement suggestions for the Use Case technique, the respondents were asked to check the suggestions that they thought would have been useful. Then the respondents were asked to write some improvement suggestions themselves. At the end of the survey the respondents were allowed to write general comments.

Chapter 7

Operation of the Interviews and the Survey

This chapter describes how we performed the interviews and surveys. Section 7.1 describes the operation of the interviews, and Section 7.2 describes the operation of the survey.

7.1 Operation of the Interviews

This section describes the preparation, execution and data validation of the interviews with the developers and the clients.

7.1.1 Preparation

In advance of the interviews we asked the interviewees, both developers and clients, to bring an example of a Use Case from the project they were working on at the time being. This was useful since we could talk about a specific Use Case that both we and the interviewee had read carefully before the interview started.

7.1.2 Execution

We performed ten interviews with software developers, where two of the interviews contained two interviewees, one contained three interviewees and the rest contained one interviewee. In total we interviewed fourteen developers. In addition, we interviewed two client representatives and these interviews took between ten and fifteen minutes. The single person interviews took about 30 minutes, while the interviews with more than one person took about an hour. All interviews, except for one, were taped. This was done with permission from the interviewees.

The interviews started with that the interviewees were informed about what the interview was about and the intention of it. In all the interviews we followed the interview guide in Appendix A to some extent, i.e we did not ask all the questions in the order they were supposed to. We tried not to focus too much on the interview guide to maintain a natural flow in the conversation. We assured ourself in the end of the interview that we had remembered to ask all of the questions we had prepared.

As for the interviews with more than one interviewee it was harder to maintain control of the conversation because the interviewees discussed the topic with each other. This had both positive and negative effects. On the negative side the conversation had a tendency to shift focus to topics not so relevant to us, but on the positive side this gave us extra insight in the opinions of the interviewees. We also noticed that the interview was useful for the interviewees as well because during the conversation they discovered new problems they had not talked about before. The interview with one of the companies was interrupted in the end, and therefore we did not talk about the last part of the interview with them, namely personal experience and improvement suggestions. This company is later referred to as Company C.

7.1.3 Data Validation

We transcribed all the interviews to make the analysis easier, and to make sure that we did not forget anything that had been said. We did not send the transcriptions back to the interviewees for approval.

7.2 Operation of the Survey

This section describes the preparation, execution and data validation of the survey.

7.2.1 Preparation

Before we sent out the survey, two persons read through the survey to reveal vagueness in the question wording. We received a list of company names from a student organization. This list was used to find contact information on the Internet. We e-mailed about sixty six companies, asking if they could reply to our survey or forward it to relevant persons in the company. It was also sent to the persons we had interviewed since the survey contained new questions. Because we asked the persons we contacted to forward the survey, we do not know the exact number of persons that received it.

7.2.2 Execution

The survey was written in Microsoft Word, so the respondents had to fill out the Word document and e-mail it back to us. We received thirty eight replies from twenty two companies. We do not know the exact time they spent on it, but we believe it took approximately ten to fifteen minutes to answer, depending on how much complementary text each person wrote.

7.2.3 Data Validation

We removed five assertions under question 4.2 in the survey because we realized that the question wording was bad. One of these assertions was: *The detailing level of the Use Cases is appropriate*. The respondents had to rate the assertion on a scale from one, *I strongly disagree*, to five, *I strongly agree*. When a respondent checked for one, *I strongly disagree*, we could not say whether he or she thought the Use Cases had too much detail, or too little detail. We could only say that the respondent did not like the level of detail. We should have formulated the assertion another way, for instance *There are too much detail in the Use Cases*. With this formulation we would have been able to say whether the respondent thought the Use Cases had too much or too little details.

Part III

Results and Discussion

Chapter 8

Results from the Interviews

This chapter presents the results from the interviews. The companies are presented as Company A, Company B, Company C and Company D, in accordance with the anonymity. Section 8.1 presents Company A, Section 8.2 presents Company B, Section 8.3 presents Company C, and Section 8.4 presents Company D. The results for Company A is somewhat different from the other companies because in Company A we focused on one particular project while in the other companies we talked about Use Cases on a general basis.

8.1 Company A

This section presents the results from the interviews in Company A.It describes personal background of the interviewees, general information about the company, how Use Cases were applied in one particular project, the results from the interview with the client representatives and last, the developers personal opinions and personal experiences with Use Cases.

8.1.1 Information about the Company

Company A delivers a variety of services, including competences on system development and consultancy, to large and middle-sized companies in Norway. In addition to these competences they deliver solutions like enterprise portals, Business Intelligence (BI), Customer Relationship Management (CRM) and custom solutions. All solutions are based on Java or Microsoft technology.

8.1.2 Application of Use Cases in one Particular Project

We interviewed seven developers from this company that all had worked in the same project where an Internet portal was developed. Table 8.1 gives key information about the project and the interviewees. The persons marked with a "*" are people hired into this project from other companies. All of the interviewees had read Use Cases, and the system responsible and the architect had also experience from writing Use Cases.

Company A's Internet portal project

Information about interviewees

Project manager*:	Three years of experience	Three projects with Use Case
System responsible*:	Eight years of experience	Three projects with Use Case
Senior consultant:	Ten years of experience	One project with Use Case
Consultant 1:	One years of experience	One project with Use Case
Consultant 2:	Nine years of experience	Five projects with Use Case
Interface programmer*:	Four years of experience	Two projects with Use Case
Architect:	Twenty years of experience	Six projects with Use Case
Client 1:	N/A	None
Client 2:	N/A	None

What was developed
An Internet portal for an insurance company

Length of project
Five to six months

Development methodology used in this project
Scrum

Employed Use Cases for
Documentation
Testing
Development
Architecture

Number of Use Case descriptions
Eighteen

Table 8.1: Company A: Overview of the Internet portal project

The project team started producing the Use Cases by making a prototype and then dividing this prototype into Use Case descriptions. The Use Case descriptions were written by one person dedicated to the work of writing and maintaining them, and this person had long experience with writing Use Case descriptions.

According to the project manager the Use Case descriptions were used for testing, documentation, development and architecture.

The Use Case template used in this project is shown in Appendix C. The business rules were excluded from the Use Case descriptions and put into separate documents. The Use Case descriptions contained normal parts like *Use Case ID*, *name*, *actors*, *pre conditions*

etc. Instead of a normal flow of behavior they contained a set of scenarios. One Use Case could contain up to twenty one scenarios, where one scenario could be about one page long. A scenario could look like the one described in Table 8.2 (the example is modified).

Scenario 1:
1. Log in 1.1 «User name» (text field) -> **Fred** 1.2 «Password» (text field) -> **12345** 1.3 «Forgotten password» (button) 1.4 «Log in» (button) -> **Choose this** 2. Etc.

Table 8.2: Company A: Scenario example

The scenarios were usually more detailed than the one in Table 8.2, and one step could expand to five levels, for example step 1.2.2.4.5. As seen from the example above, the Use Case descriptions contained information about the user interface, for example that to log in it is required to push a button.

The project manager said that in further development of the Internet portal they would apply User stories instead of Use Cases because the Use Case description became too extensive.

8.1.3 Client Representative Interviews

In addition to interviewing the developers in this project, we interviewed two client representatives that worked in product development and with system support. They used the scenarios in the Use Cases for the purpose of testing and had no experience with Use Case descriptions from previous projects. One of the representatives had done this type of testing before, but never as structured as it was done in this project. Despite the fact that they had never used the Use Case technique before, they found them easy to use. By using the technique, one of the clients meant that it was easy to know what to do, i.e what to test, because of the high detailing level in the Use Case descriptions.

Both interviewees thought the Use Case descriptions were useful when testing. One of the interviewees said that it was useful because it gave structure to the testing. The other interviewee said that it was useful because she knew exactly what to test, and when she was finished testing. She had experienced before that she had to come up with what to test herself, and she never knew if she had run all the tests necessary.

Even though the Use Case descriptions were useful for testing, the interviewees had some comments on the layout. Both meant that some parts of the Use Case descriptions were unnecessary for their use. One of them suggested that those fields could be written in gray to be less visual because this would have made the reading of the Use Case descriptions

easier. Another thing that also contributed to the Use Case descriptions being harder to read was, according to one of the interviewees, the fact that they had to look up information in other documents when reading the Use Case descriptions. This made the reading a bit disorganized. It would have been better to have all the information in one document. The other interviewee, on the other hand, did not think of this as a problem. Another comment from one of the interviewees was that the abbreviations in the Use Case descriptions were hard to understand in the beginning.

Other comments regarding the layout of the Use Case descriptions regarded the length, numbering and level of detail. Both agreed on the layout and numbering being good, but that the Use Case descriptions were too detailed. When it came to the length of the Use Case descriptions, one interviewee meant the length was appropriate, while the other meant that they were a little bit too lengthy.

In general, both interviewees thought that Use Case description was a good thing because it gave structure to the testing.

8.1.4 Personal Opinions of Use Cases

This subsection presents the developers personal opinions about how Use Cases should be applied in projects, about who should write Use Cases, about in what projects Use Cases should be employed, about what is difficult and simple about Use Cases, about how well the Use Case technique work in discussions with clients and about Use Case guidelines.

Opinions about the Use Case Descriptions Used in the Internet Portal Project

When we asked how the interviewees thought the Use Case descriptions worked, both consultants, the architect and the project manager agreed that the Use Case descriptions worked well in the beginning of the project, but that they were too detailed. Since there were a lot of changes during the project, the work of updating and maintaining the Use Case descriptions became too time-consuming.

The system responsible stated that "This was the best project where Use Case descriptions have been used." The background for this statement was the fact that one person was dedicated to the work of writing and maintaining the Use Case descriptions. The importance of his work was shown when he left the project, and the Use Case descriptions were no longer updated, which in turn had the result that they were no longer used.

Opinions about how Use Cases Should Be Applied in Projects

There were different opinions among the interviewees on how Use Cases should be used in a project. The system responsible stated that Use Cases should be used as a base for a prototype as early as possible in the project. This prototype will make a good explanation to what shall be delivered in the project. The senior consultant thought the Use Case descriptions worked well during testing, and thought that they should be used for testing. Consultant 2 meant that the Use Case descriptions should be used with care, and only in projects where you have people dedicated to writing and maintaining the Use Case

descriptions. The comment we got from the system responsible when we asked him what purpose Use Cases should be used for, was "Use Cases should be used throughout the whole project. The alternative is not to write them."

Consultant 2 said that Use Case descriptions are useful for documentation of all tasks, in such a way that all participants agree on what should be done. He also said that Use Case descriptions are used for quality assurance of the underlying system, and to describe the functionality of the prototype. Table 8.3 summaries the opinions on how Use Cases should be applied in projects.

Summary of how Use Cases should be applied	
System responsible	As a base for a prototype
Senior consultant	For testing
Consultant 2	One or more persons should be dedicated to writing and maintaining Use Case descriptions Use Case descriptions should describe the functionality of the prototype, and document all tasks.

Table 8.3: Company A: How Use Cases should be applied

Opinions about who should Write Use Cases

We did not get many opinions about who should write Use Cases, but according to the architect, Use Case descriptions should be written by people with mathematical or technical background so that they do not get inaccurate.

Opinions about how Well Use Cases Work in Discussions with Clients

When we asked about what the interviewees thought about using Use Cases in discussions with clients, the system responsible said that clients often prefer to talk about the interface right away instead of discussing the Use Cases. His experience was that he got more valuable feedback on interface prototypes than on Use Cases. Experience about how the Use Case descriptions worked in discussions with clients in the Internet portal project is described in Section 8.1.5.

Opinions about Use Case Guidelines

The architect mentioned that he thought of Alistair Cockburn's Use Cases as the best way of writing Use Cases. He had tried the guidelines in RUP. His experience was that by following that guideline, too much documentation was created and that Alistair Cockburn's format was much more compact, see Chapter 3.

Opinions about in what Types of Projects Use Cases should Be Employed

The interviewees had several opinions on which projects Use Case descriptions should be applied. According to Consultant 2, all kinds of projects are appropriate to Use Case descriptions because you always have to have one kind of description of what the project shall deliver. The project manager and the senior consultant meant that Use Cases are appropriate in Internet portal projects and the interface programmer meant that they are appropriate in projects where new systems are developed. Consultant 2 and the system responsible meant they are appropriate in large projects. By large projects he meant projects where the creation of Use Cases do not take more then 20% of the project's time.

The senior consultant stated that Use Cases are appropriate for projects where the developers and the testers are working close or sitting within short distance. The architect meant that Use Cases are appropriate in projects where the distance between the developers and clients is large. When this distance is small, User stories are more appropriate. Table 8.4 summaries the opinions about in what types of projects Use Cases should be employed.

Summary of opinions about types of projects	
Consultant 1	All kinds of projects
Interface programmer	In projects where new systems are developed
Senior consultant	In web-portal projects
System responsible	In large projects where the creation of Use Cases takes no longer than 20% of the project's time.
Senior consultant	In projects where the developers and testers are sitting within short distance.
Architect	In projects where the distance between the developers and clients is large.

Table 8.4: Company A: Types of projects where Use Cases should be applied

Opinions about what is Difficult and Simple about Use Cases

What is difficult: All the interviewees were asked what they think is difficult regarding reading or writing Use Case descriptions. According to the system responsible, "Use Cases are not difficult to write, just boring." Consultant 2 agreed that Use Cases could be boring to read, at least when the Use Case descriptions are long. He also said that if the Use Cases become too long, the reader has a tendency to get caught up in details rather than in the high level flow. Long Use Cases also makes it difficult to get an overview of the Use Case. Table 8.5 summaries what the interviewees mentioned as difficult.

Summary of what is difficult	
System responsible	Use Cases are not difficult, just boring.
Consultant	With long Use Cases the reader get caught up in details and looses the high level flow. It is difficult to get overview of long Use Cases.

Table 8.5: Company A: What makes Use Cases difficult

What is simple: Consultant 2 and the system responsible thought Use Cases make it easier to see which tasks the project includes and what you are supposed to do before the project starts. As opposed to those meaning that it is difficult to get the overview of Use Case descriptions, the senior consultant pointed out that the Use Case descriptions are easy to follow. While talking about User stories contra Use Case descriptions, Consultant 1 stated that Use Case descriptions are more precise and less ambiguous than User stories. Table 8.6 summaries what the interviewees mentioned as easy.

Summary of what is easy	
Consultant 2	Make it easier to see which tasks the project includes and what you are supposed to do.
Senior consultant	Use Case descriptions are easy to follow.
Consultant 1	Use Case descriptions are more precise than User stories.

Table 8.6: Company A: What makes Use Cases easy

8.1.5 Personal Experiences with Use Cases in the Internet Portal Project

All the interviewees were asked to mention some positive and negative experiences with the Use Case descriptions, and to come up with some improvement suggestions for the technique.

Positive Experience

The project manager mentioned the fact that they had one person dedicated to the work of writing and maintaining the Use Case descriptions as a positive experience. She had never managed a project that was as effective and with so few errors as in this project. Consultant 2 mentioned some general positive experience: Use Cases work well as an introduction to a system you are unfamiliar with, and that it is useful that all the tasks are written down. He meant that this makes it possible to avoid discussions and conflicts because everyone knows what to do.

Negative Experience

Several of the interviewees mentioned the level of detail on the Use Case descriptions as a negative experience. The project manager mentioned the fact that some of the Use Case descriptions were fifteen to sixteen pages long as one problem, because this resulted in that the updating of the Use Case descriptions became too time-consuming. According to Consultant 2, the detailing level of these Use Case descriptions led to that some of the intended readers did not read them carefully enough. The interface programmer meant that the Use Case descriptions contained too much prose and mentioned this as a negative experience because this made it harder to read them. On the other hand, the system responsible meant that it was appropriate with the prose.

Another negative experience was that the Use Case descriptions did not work in communication with the client. This was mentioned both by the architect and the project manager. The reason was that the Use Case descriptions were too abstract and technical. "Non-technical people will have difficult understanding them (Use Case descriptions)", according to the architect. The project manager suggested to use a prototype to communicate with the client. Table 8.7 summaries the negative experiences mentioned.

Summary of negative experiences

Consultant 1 and 2, architect and *project manager*	The high level of detail
Project manager	The length of Use Cases were too long
Interface programmer	Too much prose in the Use Cases
The architect and *Project manager*	The Use Case descriptions did not work in communication with client

Table 8.7: Company A: Negative experiences

Improvement Suggestions

In connection with the negative experiences mentioned by the interviewees we asked if they had suggestions for improvements. The project manager, Consultant 2 and the interface programmer, mentioned that Use Cases should be replaced by User stories, and that this would help on many of the problems. At the same time, the architect also said that whether you use Use Case descriptions or User stories is not important, most important is that the system is described at a high level. User stories were also mentioned because they contain less prose, and would therefore be easier to use.

The interface programmer meant that it would be a good idea to decide early in the project when to stop using the Use Case descriptions. This would probably have reduced the need for updating the Use Case descriptions. In addition the interface programmer suggested to write the Use Case descriptions in a format that makes it possible for the reader to expand the parts in the Use Case descriptions according to the purpose the reader is using them. It should also be possible to easily click and jump between related Use Case descriptions and documents. Consultant 2 agreed to the suggestion of making it easier to navigate between related documents. He also requested a text editor that could make the writing of Use Cases easier.

The senior consultant and the architect suggested that a glossary or appendix with terms should be created early in the project because this would make it possible for all the participants to agree on the terms and increase the precision level of the Use Case descriptions. Table 8.8 summaries the improvement suggestions.

Summary of improvement suggestions

The project manager, *Consultant 2* and *the interface programmer*	Use Cases should be replaced by User stories
Interface programmer	One should decide early in the project when to stop using the Use Case descriptions
Interface programmer and *Consultant 2*	A tool that makes it possible to expand parts in the Use Case, and jump from Use Case to other documents.
Consultant 2	A text editor that makes it easier to write Use Cases
Senior consultant and *architect*	A glossary or appendix with terms that should be created early in the project.

Table 8.8: Company A: Improvement suggestions

8.2 Company B

This section presents the results from the interviews in Company B. It describes personal background of the interviewees, general information about the company, how Use Cases are applied in the company, and last, the developers personal opinions and personal experiences with Use Cases.

8.2.1 Information about the Company

Company B is one of Norway's largest companies in life insurance and saving for retirement. They develop systems for production and administration of portfolios. This company's end-users are private persons and companies, which interact with the systems through portals. The company has an internal information technology section that develops these systems.

8.2.2 Personal Background

In this company we got in touch with three people, one test manager and two system analysts. The development methodology used in this company is a modified version of Rational Unified Process (RUP). Table 8.9 gives a summary of personal information and information about how Use Cases are applied in the company.

Company B:
Information about interviewees and Use Cases

Interviewees
Test manager: 9 years of experience
System analyst 1: 15 years of experience
System analyst 2: 5 years of experience

Type of Use Cases applied
Primarily Use Case descriptions

Use Cases applied for
Requirements elicitation
Testing
Development

Intended readers of Use Cases
Developers
Testers

Development methodology used
Modified RUP

Table 8.9: Company B: Information about the interviewees and the Use Cases

8.2.3 Application of Use Cases in the Company

According to System analyst 1, the writing of Use Case descriptions is based on a conversation with the client. This conversation could take form as an interview or a meeting. In both cases, the client writes the business requirements and hands them over to the system analysts that use them as a base for the Use Cases descriptions. In the meetings, the work flow is drawn and each step in the Use Case description is presented and discussed with the client. Appendix C shows the template company B uses when the Use Case descriptions are written.

According to System analyst 1, this company puts more information in the Use Case descriptions than the theory describes. In addition to the actor and the normal and alternative flows, they also have references to other documents and a log of changes, among other things. According to all three interviewees, the Use Case descriptions are primarily written for the testers and developers. Those writing them are the system analysts, and three to four people write each Use Case description.

The Use Case descriptions are written they are presented to the testers and developers in a meeting. According to the test manager, it is important that the testers have the opportunity to give feedback to the Use Case descriptions as soon as possible, and then making sure that they are at an appropriate level of detail. The test manager meant that these meetings should be separated in two: one for the developers and one for the testers. The reason for this was that he thought the meetings would become more efficient this way because the developers and testers often end up in discussions about the contents of the Use Cases. Because it is hard to satisfy both testers and developers, these discussions never end.

8.2.4 Personal Opinions of Use Cases

This section presents some of the interviewees personal opinions about Use Cases.

Opinions about how Use Cases should be Applied in Projects

The test manager meant that Use Cases should be applied for the purpose of development and testing.

Opinions about who should Write the Use Cases

The system analysts meant that the Use Cases should be written by three to four system analysts.

Opinions about in what Types of Projects Use Cases should Be Employed

System analyst 1 meant that Use Case is well suited for projects that develop new systems, and not so well suited for development of already existing systems. In this last case, the Use Case writing would be a waste of time.

Opinions about what is Difficult and what is Simple about Use Cases

What is difficult: The test manager said that Use Cases that are written on many pages have the disadvantage that the reader loses the overview and gets confused. To reduce this problem, it is important that the Use Cases are structured and related to each other in a clear way. System analyst 1 mentioned that it is difficult to know what is the appropriate level of detail to write the Use Cases at.

What is simple: System analyst 2 stated that using Use Cases requires practice, but when you have learned how to use it, it is an easy way of capturing requirements. The test manager also mentioned that it is easy to use Use Cases as long as they are written in an easy-to-understand language and you have the definitions readily available.

Table 8.10 gives an overview of what the interviewees thought of as simple and difficult with Use Cases. In addition it summaries in what kind of project the interviewees thought Use Cases are appropriate.

Summary of personal opinions	
Test manager	Use Cases should be applied for testing and development.
	Long Use Case descriptions are difficult to analyze, and the reader loses the overview.
	It is easy to use Use Case descriptions when they are written in an easy-to-understand language.
System analyst 1	Use Cases are well suited for development of new systems.
	It is difficult to know what is the appropriate level of detail.
System analyst 2	Using Use Cases requires practice

Table 8.10: Company B: Personal opinions

Opinions about how Well Use Cases Work in Discussions with Clients

According to System analyst 2, Use Cases do not suit clients that are unfamiliar to Use Cases and that have not participated in the process of writing them. The same person also said "The circles are not that easy to sell" when talking about the Use Case diagrams. The test manager came up with some contradictory statements about the Use Case diagrams. He meant that the stick-men makes the diagram look simple and opens for people to think simple. The test manager also said that the Use Cases are suitable to elicit good requirements from clients with little or no experience with information technology.

The system analysts on the other side, meant that it is easier for the client to relate to interfaces than Use Case descriptions.

Summary of Use Cases in discussions with clients	
Test manager	Use Cases are suitable to elicit good requirements from clients with little or no experience with information technology.
	The stick-men makes the diagram look simple.
System analyst 2	Easier for the client to relate to a user interface prototype.
	The circles in Use Case diagrams are difficult for the client to understand.

Table 8.11: Company B: Opinions about Use Cases in discussions with clients

8.2.5 Personal Experiences with Use Cases

"I actually like Use Cases" was System analyst 2's answer when we asked her to mention some positive experiences with Use Cases. Both system analysts agreed that Use Cases are the ideal way of "attacking" the system and get good requirements. All three interviewees agreed that the Use Case descriptions can be superficial, and that is one of the reasons why it is important to let the developers and testers give feedback on the Use Case descriptions at an early stage in the project.

We asked if the interviewees had some improvement suggestions for the Use Case technique. The test manager suggested that it would have been appropriate with a standardization on the structure of content and level of detail. System analyst 1 stated that it is up to the person using the technique to apply it in a clever way. To help on the difficulties, System analyst 2 suggested using activity diagrams in addition to the Use Cases. Table 8.12 gives an overview of the interviewees positive and negative experiences with Use Cases, in addition to some improvement suggestions.

Summary of personal experiences	
Test manager	Standardization of structure and content.
System analyst 2	Activity diagrams as a remedy.
All three interviewees	Use Case descriptions can be superficial.

Table 8.12: Company B: Personal experiences with Use Cases

8.3 Company C

This section presents the results from the interviews in Company C. It describes personal background of the interviewees, general information about the company, how Use Cases are applied in the company, and last, the developers personal opinions and experience.

8.3.1 Information about the Company

Company C has customers in a variety of sectors. Network solutions, Banking and finance, and IT- consultancy are some of them. In this interview we talked to one of the sections located in Oslo. Company C's main deliveries are in two main areas, which is operations and solutions. In both these areas, Company C is one of the leading community in Scandinavia.

8.3.2 Personal Background

We interviewed two system analysts that had as their main responsibilities to write and maintain the Use Cases. The third person that participated in the meeting was a senior consulting engineer that wanted to listen to what experiences the system analysts had with the way they are using Use Case descriptions in this company. Table 8.13 gives a summary of personal information and how Use Cases are applied in the company.

Company C: **Information about interviewees and Use Cases**
Interviewees
System analyst 1
System analyst 2
Senior consulting engineer
Type of Use Cases applied
Mainly Use Case descriptions
Use Cases applied for
Requirement elicitation
Estimation
Creating test cases
Programming
Writing user guidelines
Intended readers
Client and user, and developers and testers
Development methodology used
A modification of RUP

Table 8.13: Company C: Information about the interviewees and the Use Cases

8.3.3 Application of Use Cases in the Company

The Use Case descriptions in this company are used for requirement elicitation, estimation, creating test cases, development, and writing user guidelines.

To the question *Why are you applying Use Cases?* the system analysts pointed out that Use Cases are a structured way of specifying requirements. This company uses the guidelines from RUP when writing Use Case descriptions. When a project is started, they write an example of how the Use Case descriptions shall look. Therefore the design of the Use Case descriptions could vary from one project to another, but they are often similar. The Use Case descriptions are no longer than seven pages, including the front page. A template of company C's Use Case description can be found in Appendix C. Each Use Case includes one main flow and maximum three to four alternative flows.

When making Use Case descriptions they start with a survey or a rough prototype, and then the Use Case descriptions are elaborated. To get an overview of the system, a Use Case diagram is sometimes written. This diagram is further divided into Use Case descriptions. According to the RUP guideline, the Use Case specifications should contain both non-functional and functional requirements, but System analyst 1 said that they have decided to keep the non-functional requirements in a separate document that they call supplementary specification.

The Use Case descriptions are mainly written for the client and user of the system. In addition, the system analysts thought that the Use Cases should be carefully read by testers and developers to make sure that the system satisfies the requested system. They also said that it is important that the Use Case descriptions are not written in a too technical way, to ensure that the client understands the Use Cases. To make sure that both the testers and developers are satisfied with the Use Case descriptions, the system analysts arrange meetings where the Use Case descriptions are presented for the developers and testers.

8.3.4 Personal Opinions of Use Cases

This section presents some of the interviewees personal opinions about Use Cases.

Opinions about who should Write the Use Cases

All three interviewees thought it is important that two or more persons write each Use Case, because then you get more viewpoints on the content and layout.

Opinions about Use Cases in Discussions with Clients

According to system analyst 1, it is important to go through the Use Case technique together with the client if the client has never used this technique before. This reduces the risk that the client does not understand the Use Case descriptions. In addition to this, she said it is an advantage to use a working prototype of the user interface in parallel with the Use Case descriptions, because this will make it easier for the client to see what the system is going to look like.

Opinions about how Use Cases should Be Applied in projects

System analyst 2 also pointed out that they have focused too much on Use Case descriptions in the beginning of projects, and focused less on other parts like the architecture. This has resulted in that they have lost the overall view of the project. According to the senior consultant, Use Case descriptions should not be used when existing systems are specified, but that they are more appropriate in the development of new systems.

Opinions about what is Difficult and Simple about Use Cases

System analyst 2 said that there are no big difficulties with Use Cases, but that it was a challenge to use the technique in the beginning because it required to think in a new and unfamiliar way. The biggest challenge, according to one of the system analysts, was to make it short, and not write too much prose.

General Opinions about Use Cases

System analyst 1 said "It is a challenge to get people to write Use Case descriptions, but as they get started, they see the advantage of using Use Case descriptions." According to System analyst 2, the reason for this is that the writing of Use Case descriptions requires to think in a new and unfamiliar way. The system analysts prefer to write many small Use Case descriptions rather than one long Use Case description because this makes it easier to distribute tasks to the developers. Table 8.14 gives a summary of the personal opinions.

Summary of personal opinions	
System analyst 1	Go through the Use Case technique with the client, and use a working user interface prototype in parallel with the Use Case descriptions. It is a challenge to get people to write Use Cases. In the beginning: difficult to not write too much prose.
System analyst 2	They focus too much on the Use Case descriptions in the beginning of projects. Use Case writing is easy, but it required to think in a new and unfamiliar way.
Senior consulting engineer	Use Case descriptions are most appropriate in development of new systems.
All three interviewees	Two or more persons should write each Use Case.

Table 8.14: Company C: Personal opinions

8.3.5 Personal Experiences with Use Cases

This section presents the interviewees negative and positive experience with Use Cases.

Negative Experiences

System analyst 1 mentioned that during the requirements specification phase some of the developers kept asking for Use Case descriptions. As a result, the system analysts gave the developers Use Case descriptions that were not completed. The system analysts then had to know who got which version of the Use Case description, and to tell the developers when there were changes in the unfinished Use Case descriptions. The system analysts had also experienced that testers and developers have not read the Use Case descriptions carefully enough because they seemed to be simple.

Positive Experiences

The system analysts mentioned as a positive experience that they have received positive feedback from both testers and developers that the Use Case descriptions are well written because they are easy to follow. It was also a positive experience that the developers program directly from the Use Case descriptions instead of making up their own solutions to problems. According to the system analysts this helps the project deliver a system that takes care of the client's needs and keeps the client in focus. Another positive experience with Use Case descriptions was that it is a organized way to specify requirements, and that it works well in discussions with clients. Table 8.15 summaries the personal experience of the interviewees.

	Summary of personal experience
System analyst 1	The developers kept asking for Use Case descriptions during the specification phase.
System analyst 2	Use Case descriptions seemed too simple to some testers and developers.
Both system analysts	The Use Case descriptions are easy to follow.
	Use Case description is an organized way of specifying requirements.
	Use Case descriptions gives good communication with the client.

Table 8.15: Company C: Personal experience

8.4 Company D

This section presents the results from the interview in Company D. It describes personal background of the interviewee, general information about the company, how Use Cases are applied in the company, and last, the interviewees' personal opinions and personal experience with Use Cases.

8.4.1 Information about the Company

Company D is organized as a network of independent companies in Europe and Asia. Company D offers several services in communication- and area planning. Programming, consultant services and arranging courses within design, construction and maintenance of systems are among the services offered by Company D. RUP is the developing methodology that the company is using today. The company uses both Use Case descriptions and Use Case diagrams.

8.4.2 Personal Background

We got in contact with one representative from this company. He has worked in this company since the beginning in 1988, and works as a project manager. He has experience from both reading and writing Use Cases. He has participated in several projects that has applied Use Cases and two of these applied Use Cases in presentation to the client. In addition he has used Use Cases to understand the problem to be solved in the project and to structure the project. Another interesting purpose he has used Use Cases for is organizing the company. Because Use Case is a tool to get structure, it works for both structuring the work of a computer as well as the work of an employee, according to the interviewee. Table 8.16 gives a summary of personal information and how Use Cases are applied in the company.

Company D:
Information about interviewee and Use Cases

Interviewee
Project manager - 19 years of experience

Type of Use Cases applied
Use Case descriptions
Use Case diagrams

Former use of Use Cases
Structure the project
Organizing the company
Understanding the problem to be solved

Current use of Use Cases
Analyzing the problem
Testing
Structuring the project

Methodology used
RUP

Table 8.16: Company D: Information about the interviewee and the Use Cases

8.4.3 Application of Use Cases in the Company

Use Cases are mainly used for analyzing the problem at the beginning of the project, and to communicate the tasks which the project consists of to the developers. According to the interviewee, Use Cases could also be used for testing, because the testers will get a good overview of what the system is supposed to do.

The interviewee said that the Use Cases often are written based on the requirement specification and that there are no particular role dedicated to do the writing. Just one person are writing each Use Case and the author could be a developer or a tester, among others. An example of how the Use Case descriptions were written in the interviewees last project can be found in Appendix C. The Use Case descriptions in this project were used for the purpose of getting a structure on the project. Based on the Use Case descriptions a project plan was created.

8.4.4 Personal Opinions and Experience with Use Cases

To the question *How do you think Use Cases should be applied?* the interviewee stated "They should be used when you are in need of structure". Later on he stated that "Use Cases are a tool for understanding", by which he meant that Use Cases help developers, and others to understand the totality of the system and to get the overview of the workflow.

When it comes to difficulties with Use Cases, the main difficulty is to know on which level to write the Use Cases. In addition, he has experienced that the client had trouble understanding the importance of applying Use Cases. Use Cases seemed to the interviewee to be too theoretical for some clients, and the clients therefore prefer to talk about interfaces right away. The interviewee stated that "The ideal way of applying Use Cases would be to combine the use of Use Cases inwards in the company, with the use of interface or prototypes to adjust to the client's needs". Table 8.17 summaries the interviewee's personal opinions and experience with Use Cases.

Personal opinions and experience
Use Cases should be used when you are in need of structure.
Difficult to know on which level to write the Use Cases.
The client has trouble understanding the importance of applying Use Cases.
Use Cases seem to be too theoretical for some clients.

Table 8.17: Company D: Personal opinions and experience

Chapter 9

Results from the Survey

This chapter presents the results from the survey. The results are presented according to the research question where they belong. Section 9.1 presents general information about the respondents and the Use Cases, Section 9.2 presents the result for RQ1, Section 9.3 presents the result for RQ2, Section 9.4 presents the result for RQ3, Section 9.5 presents the result for RQ4, Section 9.6 presents the result for RQ5 and Section 9.7 presents the results for RQ6. The last section, Section 9.8 summaries the results.

9.1 General Information

Throughout the survey we got general information about Use Cases and the subjects that does not fit into our research questions, but that we still think of as useful. This information is presented in this section.

9.1.1 General Information about Use Cases

The respondents were asked whether they write Use Case diagrams, Use Case descriptions or a combination of both and Figure 9.1 shows the distribution of answers. As seen from the figure a combination is most common.

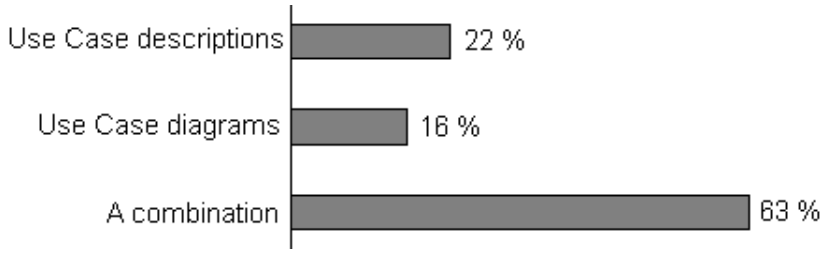

Figure 9.1: Types of Use Cases applied

The most common elements in the Use Cases were actors, main flow, alternative flows and post- and pre-conditions. It turned out that few used a log of changes.

The length of the Use Cases were between one and five pages in average, and were most commonly written in Microsoft Word and Microsoft Visio. Few of the participating companies use particular tools for organizing Use Cases. When it comes to relating Use Cases to other documents, half of the respondents did and the other half did not. About half of the respondents use guidelines for how Use Cases should be written, and many of them used the guideline in RUP.

All participating companies seemed to have similar procedures for writing Use Cases: They arrange meetings or workshops with the client or the users where the primary goal is to find the actors and goals. Based on this information an overall Use Case diagram is written and from this diagram all Use Case descriptions are identified and prioritized. When writing a Use Case description the main flow and alternative flows are described first. After that, it is common to work iteratively to finish the Use Case descriptions. The whole process take the form of a discussion between the client and the developing team, and all the details and ideas are written down. To get the right level of detail for all the intended readers, some companies have arranged internal consultative rounds together with the developers and testers. This gives them the opportunity to give feedback on the Use Case to see if they are according to their expectations and needs.

9.1.2 General Information about the Respondents

Most of the respondents worked as system developers, project managers, and consultants. They had experience from six months to twenty five years, with an average of ten years. 55% of the respondents had applied Use Cases in one to four projects, 37% in five to nine projects, and 8% in ten or more projects. Most of the respondents had written the Use Cases (82%) or used them as basis for the programming (61%), while a minor part had used the technique for testing (32%) and creation of test cases (32%).

9.2 RQ1 When is it Appropriate to Apply Use Cases?

This section presents the results for RQ1. It presents what development methodologies the companies use, what types of projects the respondents thought Use Cases are appropriate, when it is *not* appropriate to apply Use Cases and how Use Cases and User stories can be combined with other remedies.

9.2.1 Development Methodologies Used in the Companies

To see when it is appropriate to apply Use Cases, the types of development methodologies the companies used are also presented. The most common system development methodology was the RUP, and then, in increasing order, Scrum, XP and the Waterfall model. The rest said they use some form of agile methodology. A description of these methodologies can be found in Section 2.1.

9.2.2 Appropriate Projects to Apply Use Cases

It was hard to get clear, precise answers to the question about in what projects Use Cases are appropriate to use, during the interviews. The answers were quite superficial: Use Cases are appropriate to apply in large projects, web projects and in projects where completely new systems are developed. Based on these answers, a list of project types was made in the survey and the respondents had to rate them on a scale from one, *I strongly disagree*, to five, *I strongly agree.* Figure 9.2, Figure 9.3 and Figure 9.4 presents the results.

Assertion RQ1.1: *It is appropriate to apply Use Cases when developing completely new systems.*

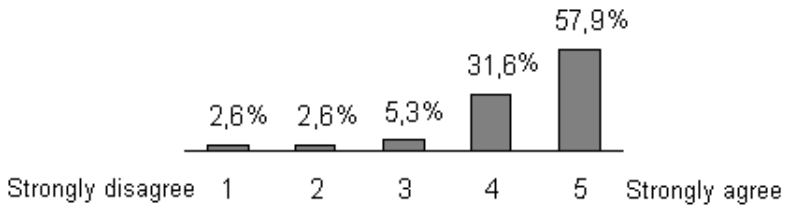

Figure 9.2: Assertion RQ1.1

Assertion RQ1.2: *It is appropriate to apply Use Cases when we develop Internet portals.*

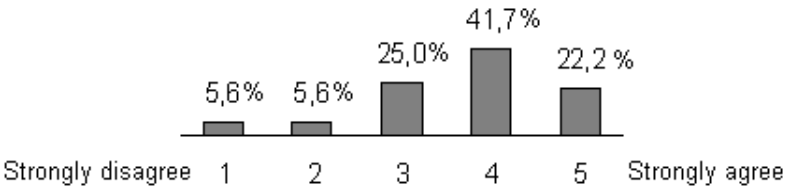

Figure 9.3: Assertion RQ1.2

Assertion RQ1.3: *It is appropriate to apply Use Cases when we develop further on existing systems.*

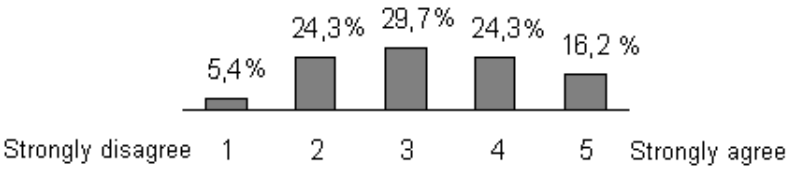

Figure 9.4: Assertion RQ1.3

9.2.3 When is it *not* Appropriate to Apply Use Cases?

One interviewee said "If you have interface prototypes you do not need Use Cases". We added this comment in the survey, and the result was an average of 1.6 which means that 58% strongly disagreed to this assertion. Only one person strongly agreed, and one person partly agreed. The result can be found in Appendix D.

The respondents to the survey were also asked if they preferred other techniques than Use Cases. The results are shown in Table 9.1 with the percentage of respondents that preferred each technique. The rest of the respondents, 73.7%, preferred Use Cases, but it is important to notice that it is uncertain how many of the respondents that have applied other techniques than Use Cases. The respondents were never asked for what requirements management techniques they had tried.

Other preferred techniques	
User stories	21.1%
Work flow diagrams	2.6%
Activity, sequence and flow diagrams	2.6%

Table 9.1: Other preferred techniques

One of the persons that preferred User stories said that she prefers short descriptions, and that a Use Case often contains too much detail that gets irrelevant or incorrect as time goes by. This results in a lot of extra work and confusion. Another person said that User stories give more freedom to the programmer, and that they spend less time on making and maintaining the requirement specification when applying User stories. Another said that in agile development with small iterations that are no longer than two weeks, it is better to apply User stories than Use Cases.

The person that preferred activity diagrams, sequence diagrams and flow diagrams reasoned that these were necessary to describe the system logic. The person that preferred work flow diagrams said the reason was that they fitted their way of making work-process support systems better than Use Cases. As a total, 26% of the respondents preferred other techniques than Use Cases.

9.2.4 Use Cases and User Stories Combined with other Remedies

To the question of what technique was preferred, some commented that they preferred to use the technique in combination with other remedies. These are presented in Table 9.2.

Combination of techniques
Use Cases together with: - User interface prototypes with descriptions. - Only user interface prototypes. - A detailed requirements specification. User stories together with: - Oral communication and code. - Interface prototypes. - A combination of user interface prototypes and test descriptions.

Table 9.2: Combination of techniques

9.3 RQ2 For What Purposes Are Use Cases Applied?

The section is divided into four parts: what Use Cases *are* used for, what developers think Use Cases *should be* used for, what developers think Use Cases *should not be* used for, and for whom they are written.

9.3.1 For what Purposes Are Use Cases Applied?

Table 9.3 shows what Use Cases are used for in the participating companies. The purposes are listed in descending order in accordance to the percentage of answers.

Use Cases *are* used for	
86%	Structure the requirements specification
86%	Estimation
82%	Programming
68%	Creating test cases
50%	System documentation
37%	Writing user documentation
34%	Client
18%	Preparing training courses
9%	Test directly from Use Cases

Table 9.3: What Use Cases are used for

As seen from Figure 9.3 Use Cases are most commonly used to structure the requirements specification, estimation and programming.

9.3.2 For what Purposes Should Use Cases Be Applied?

Table 9.4 shows for what purposes the respondents thought Use Cases should be used.

	Use Cases *should* be used for
84%	Discussions with clients
82%	Elicit good requirements
66%	Get an introduction to a unknown system
55%	Structure the project
47%	Estimate the project's costs
16%	Write users guidelines
8%	Make a training course

Table 9.4: What Use Cases should be used for

According to the respondents' opinions, Use Cases are useful for communication with the client and to elicit requirements. Writing user guidelines, and making training courses are less useful purposes for Use Cases.

9.3.3 For what Purposes Should Use Cases *Not* Be Applied?

According to what purposes Use Cases are used, we also asked the respondents for what purposes they think Use Cases should *not* be used. Table 9.5 shows the result.

	Use Cases *should not* be used for
48%	Writing user documentation
48%	Preparing training courses
43%	Test directly from Use Cases
35%	Estimation
35%	System documentation
26%	Programming
13%	Structure the requirements specification
13%	Creating test cases

Table 9.5: What Use Cases should not be used for

Writing user documentation, preparing training courses, and testing came out as the purposes that most of respondents thought of as not suitable for Use Cases. It is worth mentioning that some of the respondents thought Use Cases should not be used for the same purpose as they use it for in their projects. This result can be seen by comparing the

results in Table 9.3 and Table 9.5. This was the case in four of the purposes: For *Structuring the requirements specification*, two of the three respondents that applied Use Cases for structuring the requirements specification thought Use Cases are not appropriate for structuring requirements.

Programming was the second alternative where three of the six respondents that apply Use Cases for programming thought Use Cases are not appropriate for programming.

In the third purpose, *Estimation*, four of the eight respondents that apply Use Cases for estimation thought Use Cases are not appropriate for estimation.

For the last purpose, *System documentation*, three of the eight respondents that applied Use Cases for system documentation thought Use Cases are not appropriate for system documentation.

We cannot present the reason for why the subjects thought it was not appropriate to use Use Cases the way they are using them today, because they were not asked to reason for their opinions about this.

9.3.4 For whom Are the Use Cases Written, and who Writes them?

The results show that in some projects the writing of Use Cases is a cooperation between roles, while in other projects one person has the responsibility of writing them. The most common role of the person(s) writing Use Cases is the developer, but it is worth noticing that 58% thought that Use Cases should be written by a specialist, either in cooperation with others, or alone. In addition, 95% of the respondents thought it is an advantage that more than one person write the Use Cases. Table 9.6 shows for whom Use Cases primarily are written.

Use Cases are written for the	
53%	Developer
26%	Client
11%	Developer and client
3%	Developers and testers
3%	Collaborating partner
3%	Everyone in the project team

Table 9.6: Whom Use Cases are written for

According to Table 9.6, Use Cases are primarily written for the developers and the client.

9.4 RQ3 How Well do Use Cases Work in a Specific Project?

This research question concerns one particular project from which we interviewed developers, and has therefore relevance to the interview results. Section 8.1 presents the interview results from this project in Company A.

9.5 RQ4 Do Use Cases Work Well in Discussions with Clients?

This section presents the results for the seven assertions about Use Cases in discussions with clients and other general comments about Use Cases in discussions with clients.

9.5.1 Assertions about Use Cases in Discussions with Clients

Based on comments from the interviews, we made a set of assertions that the respondents had to make up their opinions about. They had to give their answers on a scale from one, *I strongly disagree*, to five, *I strongly agree*, and where three means *neutral*. Figure 9.5, Figure 9.6, Figure 9.7, Figure 9.8, Figure 9.9, Figure 9.10 and Figure 9.11 present the results

Assertion RQ4.1: *A prototype of interfaces should be used in addition to Use Cases, because the client needs a picture of what the system is going to look like.*

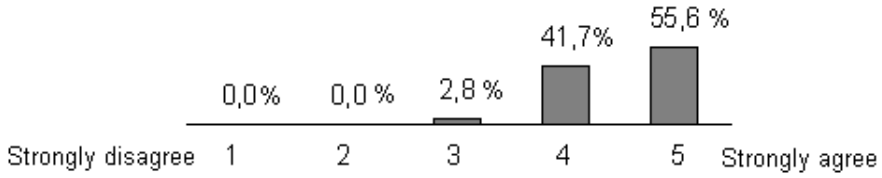

Figure 9.5: Assertion RQ4.1

Assertion RQ4.2: *I think Use Case is a good technique for communication with clients that are not so familiar with IT-technology.*

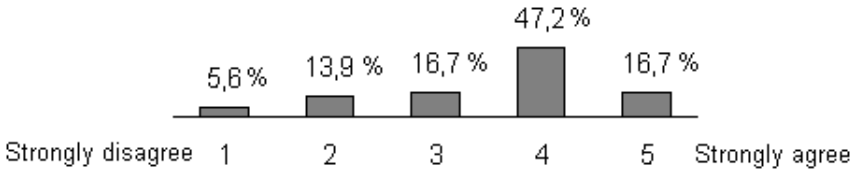

Figure 9.6: Assertion RQ4.2

Assertion RQ4.3: *I have received feedback from clients that they would rather talk about the interface than Use Cases.*

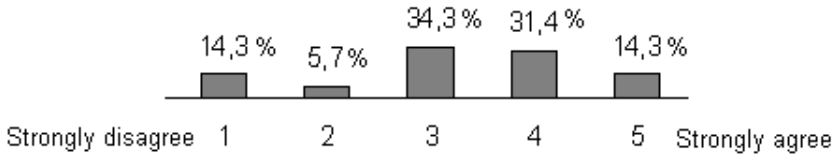

Figure 9.7: Assertion RQ4.3

Assertion RQ4.4: *Use Cases by themselves work well as a tool for communication with the client.*

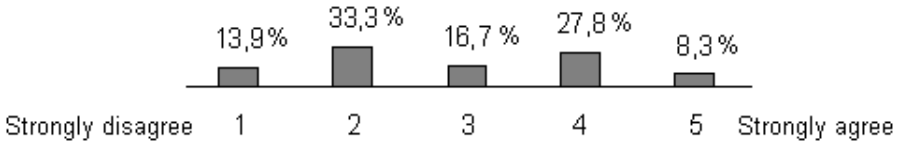

Figure 9.8: Assertion RQ4.4

Assertion RQ4.5: *I have received feedback from clients that they think it is difficult to get the overview of the Use Cases.*

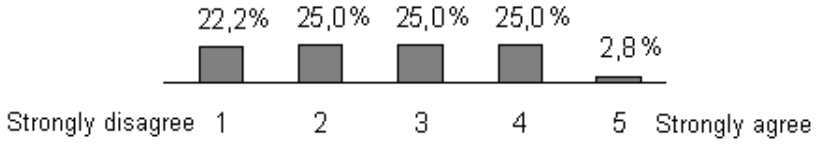

Figure 9.9: Assertion RQ4.5

Assertion RQ4.6: *The ideal solution is to apply Use Cases as a communication tool between developers, and apply User interface prototypes when talking to the client.*

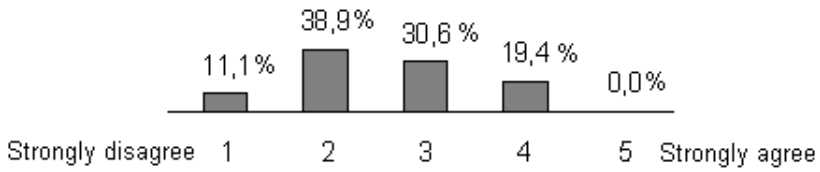

Figure 9.10: Assertion RQ4.6

Assertion RQ4.7: *I have received feedback from clients that they think Use Cases are futile.*

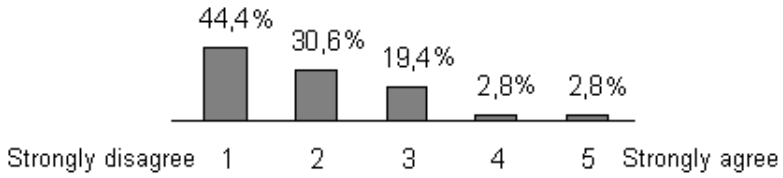

Figure 9.11: Assertion RQ4.7

9.5.2 Other Comments about Use Cases in Discussions with Clients

One respondent to the survey commented that his most important positive experience with Use Cases was as a form for communication with the client where they could see and describe the system and its behavior from the users perspective. He also meant that Use Cases work well as a communication tool because Use Cases cause the developers and the clients to compromise and give an understanding between the developer's and client's special fields. Another developer wrote that Use Case is a universal technique that works with sub suppliers in, for instance, India. Another respondent commented that Use Cases only work well as a tool for communication when it is part of a bigger set of communication resources.

The respondents had contradictory meanings about how Use Cases work as a tool for communication. For instance, one respondent wrote: "[Use Case is a] nice way to communicate with users and clients that do not have a good technical understanding. Use Cases are easy to understand with little training." While another respondent wrote: "It takes time to learn how to write and to read Use Cases".

9.6 RQ5 What is Difficult and what is Simple about Use Cases?

Based on the interviews we made a set of assertions regarding difficulties with Use Cases. Figure 9.12, Figure 9.13, Figure 9.14, Figure 9.15, Figure 9.16, Figure 9.17 and Figure 9.18 present the result. The assertions are presented in decreasing order with the one with the highest average score on top.

Assertion RQ5.1: *I think it is difficult to find the right level of detail that suits developers, testers and the client.*

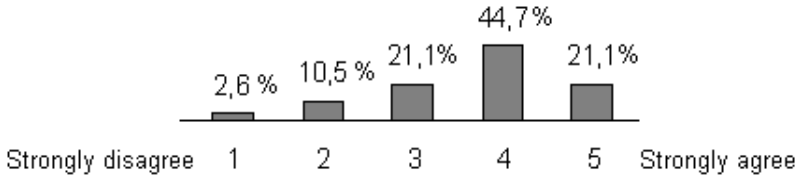

Figure 9.12: Assertion RQ5.1

Assertion RQ5.2: *I think it is difficult to keep the Use Cases updated throughout the project.*

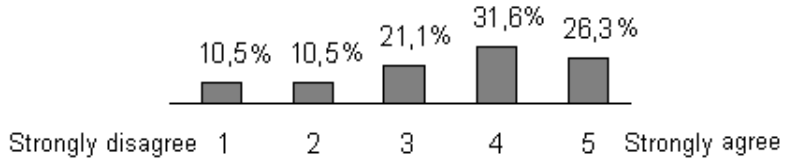

Figure 9.13: Assertion RQ5.2

Assertion RQ5.3: *I think it is difficult to keep track of changes in Use Cases.*

Figure 9.14: Assertion RQ5.3

Assertion RQ5.4: *I think it is difficult to communicate changes in the Use Cases to testers and developers.*

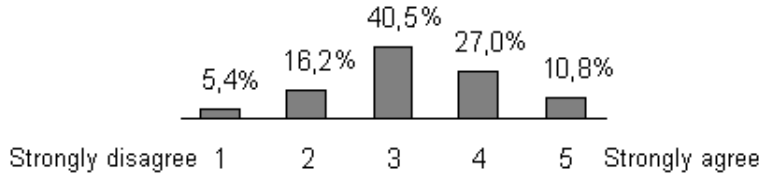

Figure 9.15: Assertion RQ5.4

Assertion RQ5.5: *I think it is difficult to prioritize Use Cases.*

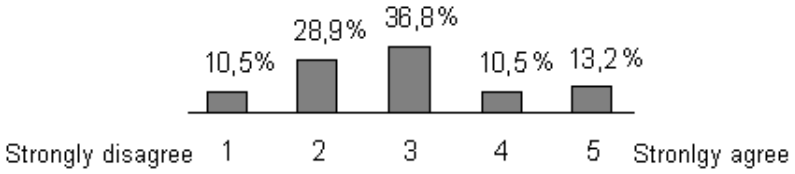

Figure 9.16: Assertion RQ5.5

Assertion RQ5.6: *I think it is difficult to see the connection between related Use Cases.*

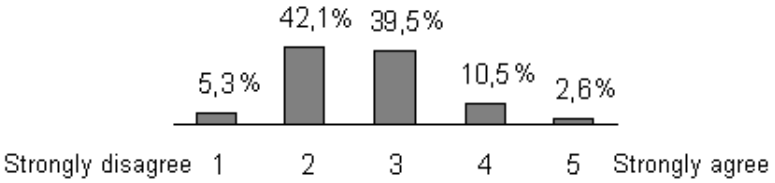

Figure 9.17: Assertion RQ5.6

Assertion RQ5.7: *I think it is difficult to write Use Cases.*

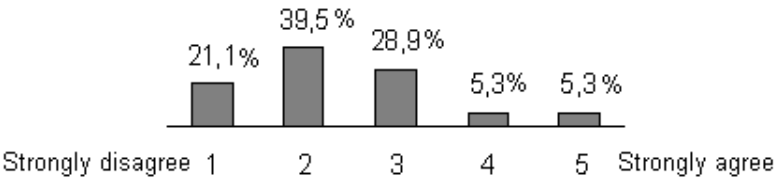

Figure 9.18: Assertion RQ5.7

9.7 RQ6 How Can we Improve the Use Case Technique?

This section presents the respondents negative experiences and improvement suggestions.

9.7.1 Negative Experiences

Based on negative experience mentioned during the interviews we made a set of assertions and the respondents had to answer either *Never experienced*, *Experienced once* or *Experienced two or more times*. The results are shown in Table 9.7, where the number of respondents that had experienced the assertion is shown in percentage.

	Negative experiences
89%	I have experienced that the Use Cases are not precise.
69%	I have experienced that Use Cases that we have used a lot of time in the beginning of the project, have become irrelevant when the implementation starts.
57%	I have experienced that Use Cases should not contain details about the user interface.
53%	I have experienced that if one person has full responsibility for the Use Cases, they are better maintained, and easier to deal with.
53%	I have experienced that Use Cases have been so extensive that I have not bothered to read them through.
33%	I have experienced that we sometimes focus too much on the Use Cases and forget other parts of the specification.
31%	I have experienced that the work with Use Cases has taken so much time that the project's progress has been prevented.
26%	I have experienced that programmers have been programming based on irrelevant Use Cases.

Table 9.7: Negative experiences with Use Cases

According to Table 9.7, the most evident experiences are that the Use Cases are not precise, and that Use Cases have become irrelevant when the implementation starts.

One of the interviewees mentioned that "Working with Use Cases is boring" as a negative experience. On a scale from one, *I strongly disagree*, to five, *I strongly agree*, the average of the respondents to the survey answered 2.26, which indicates that they partly disagree.

9.7.2 Improvements

According to the negative experiences we asked if the interviewees had done any effort to reduce these negative experiences. Few had made any effort, but some had started to use User stories instead of Use Cases. When we asked if they had any suggestions for improvements of the Use Case technique the interviewees came up with some suggestions which we used as a basis for alternatives when we asked the same question in the survey. The results from the survey is shown in Table 9.8. The improvements are ranked according to the percentage of the respondents to the alternatives.

	Possible improvements
71%	A tool that makes it easier to get an overview of related Use Cases. For example a Wiki where interfaces, business rules etc, are easily related.
63%	A standardisation of layout and content in the Use Cases.
58%	A glossary with terms that is used in a consistent way.
39%	A tool that tells when essential changes in the Use Cases are made
26%	A tool that makes it possible to expand the steps, or parts of the Use Cases, when needed.

Table 9.8: Possible improvements

Rules or guidelines for what is the most appropriate level of detail was also a suggestion that came up. This would make it easier to get the right level of detail in the Use Cases. Some also suggested that the Use Cases should not contain details about the interface.

9.8 Summary of Results

Table 9.9 presents the most important results for each research question.

	RQ1 When is it appropriate to apply Use Cases?
89%	agreed that it is appropriate to apply Use Cases when developing completely new systems.
64%	agreed it is appropriate to apply Use Cases when developing Internet portals.
	RQ2 For what purposes are Use Cases applied?
86%	apply Use Cases for structuring the requirements specification.
86%	use it for estimation.
82%	use it for programming.
68%	use it for creating test cases.
	RQ3 How well did Use Cases work in a specific project?
	See Section 8.1.2.
	RQ4 Do Use Cases work well in discussions with clients?
97%	agreed that a protoype of interfaces should be used in addition to Use Cases, because the client needs a picture of what the system is going to look like.
64%	agreed that Use Case is a good technique for communication with clients that are not so familiar with IT-technology.
75%	*disagreed* to the assertion that clients think Use Cases are futile.
	RQ5 What is difficult and what is simple about Use Cases?
66%	agreed that it is difficult to find the right level of detail that suits developers, tester and the client.
61%	*disagreed* that it is difficult to write Use Cases.
	RQ6 How can we improve the Use Case technique?
71%	agreed that it would be useful with a tool that makes it easier to get an overview of related Use Cases.
63%	agreed that it would be useful with a standardization of layout and content in the Use Cases.

Table 9.9: Summary of results

Chapter 10

Discussion of Results

This chapter discusses the results that we got from our interviews and surveys and relates them to previous research. Based on the results we conclude with a list of suggestions of how the Use Case technique can be improved.

RQ1 When is it appropriate to apply Use Cases?

Result: When developing completely new systems and Internet portals.

Discussion of result: According to our results, Use Cases are appropriate to employ when developing completely new systems, and not that appropriate when developing further on existing systems. The reason for this is that for existing systems much of the specification is already written down and making Use Cases would be superfluous.

Use Cases are also appropriate to use when developing Internet portal systems. There were not given any reasons for this choice. There are no previous research on this topic, so we cannot relate our results to other researcher's work.

RQ2 For what purposes are Use Cases applied?

Result: For structuring the requirements specification, estimating, programming and constructing test cases. Less commonly also used for system documentation, writing user documentation, preparing courses and testing directly from Use Cases.

Discussion of results: The results show that most of the companies employ Use Cases for structuring the requirements specification and for estimating projects. Many also use it for programming and to create test cases. This is in accordance with the theory of how Use Cases should be used (see Section 3.10).

RQ3 How well did Use Cases work in a specific project?

Result: For the purpose of testing from the scenarios in the Use Case descriptions, the testers thought the Use Case descriptions worked well, but the Use Case descriptions became long and too detailed according to the developers.

Discussion of result: We got the impression that this way of applying Use Cases worked well in this project. The project leader stated that she was pleased with the implementation of this particular project. The reason for the success was that one person had full responsibility for the Use Case descriptions. The testers were also pleased with the Use Cases because they were easy to read and made it easy to know what to test. A drawback with this use was that the Use Case descriptions became long and hard to keep updated.

RQ4 Do Use Cases work well in discussions with clients?

Result: Yes, if used together with user interface prototypes and written in a language that the client understands.

Discussion of result:F The opinions on Use Cases as a tool for communication are divided. Some developers think that Use Cases should be used primarily as a tool for communication, while others believe Use Cases should not be used for this purpose at all. Either way, the most evident result of this study is the following: A prototype of the interface should be used in addition to the Use Case when you communicate with the client. This does not, however, imply that prototypes can substitute Use Cases.

Other researchers have suggested to make Use Cases more understandable by setting them into a context, i.e in what situations the Use Case is relevant. It is also recommended that the Use Cases are described in a way that matches the way the client thinks of the system, and that technical words are kept out of the Use Cases. Our findings could be combined with the other researcher's suggestion: Both use a user interface prototype and follow the recommendations for how to write Use Cases that are understandable by clients.

The results from our research also showed that many developers believe that Use Case is a good technique for communicating with clients that are not so familiar with information technology. This might indicate that the writers of the Use Cases are good at keeping technical words out of the Use Cases, as recommended in the theory. Another point worth noticing is that very few developers have received feedback from clients that they think that making Use Cases is futile. If the developers have got the right impression, clients also think it is useful to apply Use Cases. This supports the idea that Use Cases work in the communication with clients.

RQ5 What is difficult and what is easy about Use Cases?

Result: It is difficult to find the right level of detail, but beyond that it is easy to write Use Cases.

Discussion of result: The results show that it is difficult to find the right level of detail in the Use Cases that fits developers, testers and the client, but that it is not generally difficult to write Use Cases. Previous research has indicated that «include» and «extend» relations are hard to understand for both writers and readers of Use Cases. We did not get any comments on these relations in either the survey or the interview. This might indicate that companies have stopped using «extend» and «include». The impression we got from

the interviews was that most companies focus on the Use Case descriptions, and only on rare occasions write Use Case diagrams in addition.

RQ6 How can we improve the Use Case technique?

Result: Table 10.1 presents the results.

Discussion of result: There were three negative aspects about Use Cases that many of the developers had experienced: (1) that Use Cases had not been precise enough, (2) that the developers had used a lot of time in the beginning of the project on Use Cases that became irrelevant when the implementation started, and (3) that Use Cases should not contain details about the user interface.

A possible way to avoid problem number (1) would be to standardize the layout and content in the Use Cases. This could be done by making a standardization for each project or make one standardization for the whole company. Problem number (2), that Use Cases get irrelevant as time goes by, could be to not write extensive Use Cases in the beginning.

The fact that many had experienced problem number (3) is in accordance with other researcher's studies in the software industry. The previous research had shown that developers couple the Use Cases to user interface details even though it is highly recommended not to do so. Since many of the respondents to our survey had experienced problems with details about the user interface in the Use Case, our results support the rule of not inserting any information about the interface design in the Use Cases.

Based on the results from this study, we have made a list of suggestions for how to improve Use Cases in Table 10.1.

Suggestions for how to improve Use Cases	
Suggestion 1:	Make use of a tool that makes it easier to get an overview of related Use Cases and other documents.
How to:	Create a Wiki where you relate interfaces, business rules, Use Cases etc.
Suggestion 2:	Do not emphasize too much on writing extensive Use Cases in the beginning of the project.
How to:	Start with Use Case briefs or Casual form, and expand the Use Cases into fully dressed versions later in the project, see Chapter 3.
Suggestion 3:	Standardize your Use Cases.
How to:	Set clear directions for the layout and the content of the Use Cases. You should have one general template for the whole company. Each project team should adjust the template before the project starts so that it fits their needs.
Suggestion 4:	Avoid unnecessary changes in the Use Cases when the user interface design changes.
How to:	Do not write details about the user interface in the Use Cases. Instead, create a low fidelity prototype of the interface in addition to the Use Case.
Suggestion 5:	Maintain the Use Cases throughout the project.
How to:	Make sure one person has the full responsibility for updating and maintaining the Use Cases.
Suggestion 6:	Use terms in a consistent way.
How to:	Create a glossary of terms for the requirements document that is used in a consistent way.
Suggestion 7:	Make sure your client understands the Use Cases properly.
How to:	Provide a user interface prototype in addition to your Use Cases, and provide training courses for your clients.
Suggestion 8:	Get more than one point of view when writing Use Cases.
How to:	Make sure at least two persons write each Use Case.

Table 10.1: Suggestions for how to improve your Use Cases

Chapter 11

Discussion of Validity

This chapter discusses the validity of the interviews and the surveys. It is important to discuss the validity of our work to ensure that our results are trustworthy. The discussion is based on the theory of validity in (Wohlin et al., 2000).

11.1 Threats to the Interviews' Validity

First, we have never done this type of interviews before and are not experts in any way. To be sure that we got all the information we needed, we set up an interview guide that we used more or less during all the interviews. We did some stumbling in the first interviews, but as we got more used to the interview setting we became better interviewers, and better at posing follow-up questions.

Since some people are more talkative than others, we got more information from some interviewees than others. Also, some interviews were conducted with two or three interviewees and we discovered that in these interviews it was harder for us to keep to the interview guide because the interview became more as a conversation. We used the interview guide in the end of these interviews to make sure that we had received the information we needed. Despite our lack of experience with the interview setting, and personal differences between the interviewees, we believe that the information we got from the interviews are trustworthy.

One possible threat to the validity is that we have misunderstood or misinterpreted the interviews in our analysis. We did not send the analysis back to the companies for approval, so the companies had no chance of correcting the text in case we had misinterpreted something. Since we recorded and transcribed all the interviews, we believe that the chance of misinterpretation is low.

One of the interviewees felt uncomfortable with the fact that we were recording the conversation. We noticed that this person got more open and relaxed after the recorder had been shut down. We would probably have gotten more information from this person if we had not used the recorder. On the other side, it would have been more difficult to remember everything from the interview without the recorder.

The time or day the participants were interviewed may have affected the results. If, for example, a participant recently was involved in a project where the use of Use Cases was catastrophic, this person most likely focused on that particular project because he or she remembers this project best. The same applies for a person that recently has had a positive experience with Use Cases. This validity threat also applies for the survey.

One of our research questions was to investigate whether Use Cases work well in discussions with clients. Our result to this research question is based on developers opinions about this since we only got hold of two client representatives. This is a problem for the validity of the results for this question, but we believe that the developers have an idea of how well Use Cases work for this purpose, and we therefore believe that the results are trustworthy.

11.2 Threats to the Survey's Validity

Participation was voluntary and people that have strong opinions about Use Cases in a positive or negative way will be more eager to answer a survey than people that do not have strong opinions about it. Thus we will miss response from "average people". We do not believe that this has made our results less trustworthy.

The survey was written in Microsoft Word and sent via e-mail. We got feedback from one of the respondents that he would have preferred an Internet-based survey tool because they are easier to fill in. It might be that some persons did not respond to the survey because they thought it was bothersome to fill in a Word document. Also, there were some open questions in the survey that not all of the respondents bothered to answer. This has only affected our result in that way that we got less responses.

Some of the questions in the survey were answered on a scale from one, *I strongly disagree*, to five, *I strongly agree*. Since people have their own opinion of what they mean by strongly agree, and strongly disagree, this will affect the result. Some persons use the two extreme points (one and five) much, while others are more reluctant to use them. There is nothing we can do about this problem, and we do not believe this has made our result less true.

Some of the questions had alternatives, and one problem is that some of the respondents do not fall into any of the categories, and therefore check the alternative that fit best, but not perfectly. To some of these questions we added a comment field so the respondents could fill in their own answers. This made the results more reliable.

11.3 Summary of Validity

The most important threat to the validity is the fact that we have only interviewed two client representatives. This means that our results to *RQ4 Do Use Cases work well in discussions with clients?*, are primarily based on developers opinions, and not on clients opinions. Anyway, we believe our results for RQ4 are correct, but the results would have

become more trustworthy if we had interviewed more clients. To summarize, we believe that the threats to validity have been reduced to an acceptable level and that all our results are trustworthy.

Part IV

Conclusion and Further Work

Chapter 12

Conclusion

In this thesis, we have examined how project teams apply the Use Case technique by interviewing four companies and sending a survey to sixty five Norwegian companies. We have investigated when it is appropriate to apply Use Cases (RQ1), for what purposes Use Cases are applied (RQ2), how well the Use Case technique worked in an Internet portal project (RQ3), how well the Use Case technique works in discussions with clients (RQ4), and what difficulties developers and clients have run into with Use Cases (RQ5). Based on the research we have come up with a set of improvement suggestions for the Use Case technique (RQ6).

RQ1 When is it appropriate to apply Use Cases?
The results show that many developers believe that Use Cases are appropriate to use when developing new systems and Internet portal systems, but that it is not worth writing Use Cases when developing further on existing systems because most of the system is already specified.

RQ2 For what purposes are Use Cases applied?
The companies we have been in contact with apply Use Cases for structuring the requirements specification, estimation, programming and construction of test cases. They also, less commonly, apply Use Cases for system documentation, writing user documentation, preparing training courses and testing by following the scenario steps in the Use Case descriptions.

RQ3 How well did Use Cases work in a specific project?
We have particularly examined the use of Use Cases as a tool for testing by interviewing developers and testers from one project team that developed an Internet portal. This application of Use Cases worked well particularly for the testers, because the scenarios were detailed and easy to follow. The fact that the Use Cases became so detailed, made them long and hard to update and when the person with the responsibility for the Use Cases left the project, the Use Cases were no longer updated. The study of this particular project also support our believes that one person should have the dedicated responsibility of keeping Use Cases updated throughout the project because this will enhance the quality of the Use Cases.

RQ4 Do Use Cases work well in discussions with clients?
It seems that Use Cases work in discussions with clients if they are supplied with user interface prototypes and written in a language that the client understands.

RQ5 What is difficult and what is easy about Use Cases?
The most problematic aspect of Use Cases is to find an appropriate level of detail in the Use Cases that fits developers, testers and clients. We therefore suggested that developers, testers and clients should discuss and reach an agreement on the layout and contents of Use Cases before they start to write Use Cases. Other problems that many encounter are that they use a lot of time in the beginning of projects on Use Cases that become irrelevant when the implementation starts.

RQ6 How can we improve the Use Case technique?
Based on the developers experience with Use Cases we have come up with eight suggestions for how to make the Use Cases better: (1) Make use of a tool that makes it easier to get an overview of related Use Cases and other documents, (2) Do not emphasize too much on writing extensive Use Cases in the beginning of projects, (3) Standardize the Use Case template, (4) Avoid unnecessary changes in the Use Cases when the user interface design changes, (5) Maintain the Use Cases throughout the project, (6) Use terms in a consistent way, (7) Make sure the client understands the Use Cases properly and (8) Get more than one point of view when writing Use Cases.

Chapter 13

Further Work

This research has emphasized general application areas and experience with Use Cases in twenty two companies, and studied the use of Use Case in one specific project. Later research should focus more on the application of Use Cases in one particular project over an extensive period of time to get a deeper insight into how Use Cases are applied. By studying one particular project over a period of time, the researcher will get rich insight into how the application of Use Cases works, and what potential improvements exist for the company under study.

One should also emphasize more on clients to get their opinions about Use Cases because this will give rich insight into how the clients think of using Use Cases, and not only developers. This will give answers to how Use Cases should be written so that clients understand them better.

Later research should implement a tool for writing and maintaining Use Cases and related documents, and study the use of this tool in a real software project or in a student project. This will give answers to whether a tool for writing and maintaining Use Cases is helpful or not. The rest of our improvement suggestions should also be studied in a real project or a student project. This should be done to see if they are of any help to reduce the problems mentioned in this thesis.

Part V

Appendices

Appendix A

Interview Guide

This Appendix contains the questions used during the interviews. First the questions asked to the developers are presented, then the questions asked to the client are presented. The questions are written in Norwegian.

Intervju av utviklere

Presentere oss selv: Navn, skriver masteroppgave om Use Case.

Formål: Formålet med dette intervjuet er å få innsikt i hvordan din bedrift bruker Use Case, samt å få innsikt i dine erfaringer og meninger om Use Case teknikken.

Tema: Vi vil ta for oss seks deler under intervjuet:

Del I: Personlig informasjon
Del II: Kort om bedriften
Del III: Bruk av Use Case i bedriften
Del IV: Bruk av Use Case i et prosjekt
Del V: Dine meninger om Use Case
Del VI: Dine erfaringer med Use Case.

Konfidensialitet: Det som kommer fram av dette intervjuet vil bli brukt til å lage en spørreundersøkelse som skal sendes ut til bedrifter som bruker Use Case. Resultatet av intervjuet vil også bli skrevet om i rapporten. Deres navn og firmanavn vil bli sensurert dersom det er ønskelig.

Opptak: Vi ønsker å ta opp intervjuet på bånd for å gjøre det lettere for oss i ettertid. Er dette OK?

Tema: Vi vil ta for oss seks deler under intervjuet slik det foreligger på arket dere har fått: Personlig informasjon, om bedriften, Om bruk av Use Case i bedriften, om bruk av Use Case i et prosjekt, dine meninger om Use Case og til sist dine erfaringer med Use Case.

Angående spørsmålene: Dersom det er noe du ikke ønsker å svare på, må du si fra.

Spørsmål?

Del I: Personlig informasjon

1.1 Navn:

1.2 Antall år i bransjen:

1.3 Stilling:

1.4 Arbeidsoppgaver:

1.5 Hvor mange prosjekter har du vært med på som brukte Use Case?

1.6 Hvordan var du involvert med Use Casene? Skrev du, eller leste du de?

1.7 Annet:

Del II: Bedriften

2.1 Hva slags type bedrift er dette? (Konsulent, annet?)

2.2 Hva slags type kunder?

Del III: Bruk av Use Case i bedriften

3.1 Har dere retningslinjer for hvordan Use Case skal skrives og brukes i bedriften? Hvordan brukes disse?

3.2 Hva går disse retningslinjene ut på?

3.3 Use Case blir brukt til forskjellige formål, som f.eks testing eller kravdokumentasjon. Til hvilke formål blir Use Casene generelt brukt?

3.4 Hvorfor bruker dere Use Case?

3.5 Hvis dere i noen tilfeller ikke bruker Use Case, hvorfor gjør dere ikke det?

Del IV: Prosjektspesifikt

4.1 Hva slags system ble laget i prosjektet? (kunde)

4.2 Hvilken rolle har/hadde du i prosjektet?

4.3 Hvilken utviklingsmetode brukte dere? (RUP, Scrum, annet ?)

4.4 Use Case blir brukt til forskjellige formål, som f.eks testing eller kravdokumentasjon. I dette prosjektet: Til hvilket formål ble Use Casene brukt?

4.5 Hvordan fungerte denne bruken av Use Case?

Hvis ikke kravinnhenting:
4.6 Hvordan foregikk kravinnhentingen?

4.7 Hvilken rolle i det prosjektet du jobber i nå har de som skriver Use Case?

4.8 Hvor mange personer skriver hvert enkelt Use Case?

4.9 Hvilken rolle i det prosjektet du jobber i nå har de som leser Use Case?

Hvis vedkommende har skrevet Use Case:
4.10 Kan du gi eksempel på hvordan dere går fram når dere lager Use Case?

4.11 Kan du gi et eksempel på hvordan dere skriver Use Case?

4.12 Hvordan håndterer dere endringer i Use Casene?

4.13 Hvordan syns du bruken av Use Case fungerte mot kunden?

4.14 Hvorfor brukte dere Use Case?

Del V: Personlige meninger

5.1 Hvordan mener du Use Case bør brukes i prosjekter?

5.2 Syns du noe er vanskelig med å skrive Use Case?

5.3 Syns du noe er enkelt med å skrive Use Case?

5.4 Syns du noe er vanskelig med å lese Use Case?

5.5 Syns du noe er enkelt med å lese Use Case?

5.6 Syns du noe er vanskelig med bruken av Use Case generelt?

5.7 Syns du noe er enkelt med bruken av Use Case generelt?

5.8 Forslag til forbedringer av Use Case teknikken? (f.eks verktøystøtte)

5.9 Foretrekker du andre metoder enn Use Case? Hvorfor?

5.10 Hvilke typer prosjekt syns du Use Case passer bra i? (størrelse, nærhet til kunde osv.)

5.11 Hvilke typer prosjekt syns du Use Case passer mindre bra i? (størrelse, nærhet til kunde osv.)

Del VI: Personlige erfaringer

6.1 Kan du nevne noen positive erfaringer med bruk av Use Case? (Sammenheng, utviklingsmetode, begrunnelse)

6.2 Kan du nevne noen negative erfaringer med bruk av Use Case? (Sammenheng, utviklingsmetode, begrunnelse)

6.3 Har dere satt i gang tiltak for å bøte på de negative erfaringene?

6.4 Har du vært med på å bruke hjelpemidler ved siden av Use Case? Hvilke? (f.eks skjermbilde)

Intervju av kunde

Presentere oss selv: Navn, skriver masteroppgave om Use Case.

Formål: Formålet med dette intervjuet er å få innsikt i hvordan din bedrift bruker Use Case, samt å få innsikt i dine erfaringer og meninger om Use Case teknikken.

Tema: Vi vil ta for oss to deler under intervjuet:

Del I: Personlig informasjon
Del II: Bruk av Use Case

Konfidensialitet: Det som kommer fram av dette intervjuet vil bli brukt til å lage en spørreundersøkelse som skal sendes ut til bedrifter som bruker Use Case. Resultatet av intervjuet vil også bli skrevet om i rapporten. Deres navn og firmanavn vil bli sensurert dersom det er ønskelig.

Opptak: Vi ønsker å ta opp intervjuet på bånd for å gjøre det lettere for oss i ettertid. Er dette OK?

Angående spørsmålene: Dersom det er noe du ikke ønsker å svare på, må du si fra.

Spørsmål?

Del I: Personlig informasjon

1.1 Navn:

1.2 Rolle i prosjektet:

1.3 Stilling:

1.4 Brukt Use Case før:

1.5 Vært med på lignende prosjekt tidligere?

Del II: Bruk av Use Case

2.1 Hva var vanskelig å forstå med Use Casene?

2.2 Hva var enkelt å forstå med Use Casene?

2.3 Syns du det var nyttig å bruke Use Case i forbindelse med testing? Hvorfor/hvorfor ikke?

2.4 Hva syns du om kommunikasjonen med utviklerne? Noe som kunne vært gjort annerledes?

2.5 Hva syns du om lengden på Use Casene?

2.6 Hva syns du om oppsettet på Use Casene?

2.7 Hva syns du om at man må slå opp i andre Use Case for å lese et Use Case?

2.8 Hva syns du om detalj-nivået på Use Casene?

2.9 Hva syns du om nummereringen på Use Casene? Nivå?

2.10 Var det noen av feltene du syns var unødvendig for din bruk?

2.11 Tror du det hadde vært nyttig med hjelpemiddel ved siden av?

2.12 Hva syns du om Use Casene generelt?

2.13 Var det noe angående krav/Use Case du synes kunne vært gjort annerledes i prosjektet?

Appendix B

Survey

This appendix includes the survey written in Norwegian.

Spørreundersøkelse om Use Case

Masteroppgave ved NTNU
linjen for datateknikk

Margrethe Kjeøy og Gerd Stalheim

Informasjon om spørreundersøkelsen

Vi er to jenter som går siste året på datateknikk ved NTNU som skriver masteroppgave om Use Case. I den forbindelse har vi utarbeidet denne spørreundersøkelsen.

Formålet med undersøkelsen er å få innsikt i hvordan din bedrift bruker Use Case, samt å få innsikt i dine erfaringer og meninger om Use Case teknikken.

Det som kommer frem i denne undersøkelsen vil bli brukt til å skrive en rapport om måter å bruke Use Case på i ulike bedrifter. Personnavn og bedriftsnavn vil bli fjernet i rapporten.

Dersom noe er uklart i spørsmålsformuleringen, eller det er et spørsmål du ønsker å gi en grundigere forklaring på, kan dette skrives ved siden av spørsmålet. Spørreundersøkelsen sendes pr e-post til en av adressene under.

Dersom du har noen spørsmål er det bare å ta kontakt med oss på e-post:

Margrethe: margrekj@stud.ntnu.no
Gerd: gerdmelt@stud.ntnu.no

Takk for at du tar deg tid til å svare på denne undersøkelsen.

Del I: Bakgrunnsinformasjon
Denne delen tar for seg bakgrunnsinformasjon og fakta om bedriften du jobber i, og hvordan du har jobbet med Use Case.

1.1 Navn (valgfritt):_____

1.2 Navn på bedrift:_____

1.3 Stilling: _____

1.4 Antall år i bransjen:_____

1.5 Hva har du jobbet med i prosjektsammenheng?
☐ Testing
☐ Programmering
☐ Kravspesifisering
☐ Prosjektledelse
☐ Annet: _____

1.6 Hvor mange prosjekter har du vært med på som har brukt Use Case? *(Kun ett kryss)*
☐ 1 – 4
☐ 5 – 9
☐ 10 eller flere

1.7 Hvordan har du vært involvert med Use Case? *(Flere kryss tillatt)*
☐ Har skrevet Use Casene
☐ Har skrevet testcase ut i fra Use Casene
☐ Har testet ut i fra Use Casene
☐ Har programmert ut i fra Use Casene
☐ Annet: _____

1.8 Hva slags type Use Case bruker dere? *(Kun ett kryss)*
☐ Tekstlige beskrivelser
☐ UML diagram, dvs strek-menn og sirkler.
☐ Begge

1.9 Hvilken utviklingsmetode bruker dere i bedriften?
(*Kryss av den metoden som ligner mest på metoden dere bruker*).
☐ RUP
☐ XP
☐ Vannfallsmodellen
☐ Annet_____

Del II: Bruk av Use Case i bedriften
Denne delen tar for seg hvordan Use Case blir brukt i bedriften.

2.1 Hvilken rolle har de(n) som skriver Use Casene? _____

2.2 Har dere en mal for hvordan Use Case skal skrives i bedriften? *(Kun ett kryss)*
- ☐ Ja
- ☐ Ja, men vi bruker den ikke
- ☐ Nei
- ☐ Vet ikke

2.3 Hvis JA på forrige spørsmål, hvilken mal? *(Kun ett kryss)*
- ☐ RUP
- ☐ Alistair Cockburn
- ☐ Vet ikke
- ☐ Annet: _____

2.4 Hva skriver dere Use Casene i? *(Kun ett kryss)*
- ☐ Microsoft Word
- ☐ Microsoft Visio
- ☐ Rational Rose
- ☐ Annet: _____

2.5 Hvordan organiserer dere Use Case som er relatert til hverandre? *(Kun ett kryss)*
- ☐ Vi bruker ikke noe spesielt verktøy for å organisere Use Casene
- ☐ På en Wiki
- ☐ Annet: _____

2.6 Hvordan organiserer dere Use Case og relaterte dokumenter? *(Kun ett kryss)*
- ☐ Vi relaterer ikke Use Casene til andre dokumenter
- ☐ På en Wiki
- ☐ Annet: _____

2.7 Hva blir Use Case brukt til i din bedrift? *(Flere kryss tillatt).*
- ☐ Å skrive brukerdokumentasjon
- ☐ Strukturere kravspesifikasjonen
- ☐ Programmere etter
- ☐ Å lage opplæringskurs
- ☐ Estimering
- ☐ Lage test case
- ☐ Teste direkte etter Use Casene
- ☐ Systemdokumentasjon
- ☐ Annet: _____

2.8 Vi bruker disse hjelpemidlene ved siden av Use Case *(Flere kryss tillatt).*

- ☐ Skjermbildeprototyp
- ☐ Sekvensdiagram
- ☐ Aktivitetsdiagram
- ☐ Navigeringskart
- ☐ Bruker ikke hjelpemiddel
- ☐ Annet:_____

2.9 Use casene blir først og fremst skrevet for *(Kun ett kryss)*

- ☐ Oppdragsgiveren
- ☐ Utviklerne
- ☐ Testerne
- ☐ Annet:_____

2.10 Hvis du har vært med på å skrive Use Case, forklar kort hvordan dere går fram når dere lager Use Case.

2.11 Use Case beskrivelsene inneholder *(Flere kryss tillatt)*

- ☐ Forretningsregler
- ☐ Aktør
- ☐ Pre-betingelser
- ☐ Post-betingelser
- ☐ Endringslogg
- ☐ Hovedflyt
- ☐ Alternativ flyt
- ☐ Andre felter:

2.12 Våre Use Case inneholder antagelser om brukergrensesnittet. Eks: ”Trykk lagre-knapp”. *(Kun ett kryss)*

- ☐ Sant
- ☐ Usant
- ☐ Vet ikke

2.13 Hvor mange sider er en gjennomsnitts Use Case beskrivelse på? *(Kun ett kryss)*

- ☐ 1- 5
- ☐ 6 - 10
- ☐ 10 - 20
- ☐ 20 eller flere

Del III: Bruk av Use Case mot oppdragsgiver
Denne delen tar for seg erfaringer om Use Case brukt i kommunikasjon med oppdragsgiver.

3.1 *Kun ett kyss er tillatt pr. spørsmål.*

	Sterkt uenig 1	2	3	4	Sterkt enig 5
Use Case alene fungerer bra som kommunikasjonsmiddel med oppdragsgiver					
Jeg syns Use Case er en god teknikk å bruke til å kommunisere krav til oppdragsgiver når oppdragsgiver ikke er så kjent med IT-teknologi					
Skjermbildeprototyp bør brukes i tillegg til Use Case fordi oppdragsgiver er visuell av seg og trenger et bilde av hvordan systemet skal se ut					
Det ideelle er å bruke Use Case innad i prosjektet, og skjermbildeprototyp mot oppdragsgiver					
Jeg har fått tilbakemelding om at oppdragsgiver heller vil snakke om skjermbilder enn om Use Case					
Jeg har fått tilbakemelding om at oppdragsgiver syns det er bortkastet å bruke Use Case					
Jeg har fått tilbakemelding om at oppdragsgiver syns det blir vanskelig å få oversikten over Use Casene					

Del IV: Personlige meninger om Use Case
Denne delen tar for seg personlige meninger og erfaringer om Use Case.

4.1 Jeg syns det er vanskelig å
(*Kun ett kryss pr spørsmål)*

	Sterkt uenig 1	2	3	4	Sterkt enig 5
Holde Use Casene oppdaterte gjennom prosjektet					
Vite hvilket nivå man skal legge seg på når man skriver Use Case					
Holde styr på endringer i Use Case					
Se sammenhengen mellom relaterte Use Case.					
Finne rett nivå på en Use Case beskrivelse som passer både utviklere, testere og oppdragsgiver					
Få kommunisert ut endringer i Use Case til utviklere og testere					
Prioritere Use Case					

Skrive Use Case					

4.2
(*Kun ett kyss pr spørsmål)*

	Sterkt uenig 1	2	3	4	Sterkt enig 5
Detaljnivået på Use Casene er passe					
Tiden vi bruker på å skrive Use Case er passe					
Tiden vi bruker på å lese Use Case er passe					
Tiden vi bruker på å oppdatere Use Case er passe					
Vi legger passe vekt på å holde Use Casene oppdaterte					

4.3 Jeg syns Use Case passer i følgende typer prosjekt
(*Kun ett kryss er tillatt pr spørsmål)*

	Sterkt uenig 1	2	3	4	Sterkt enig 5
Nyutviklingsprosjekter					
Forvaltningsprosjekter					
Internett-portal					
I prosjekter hvor man er nær oppdragsgiver					

4.4
(*Kun ett kyss er tillatt pr spørsmål)*

	Sterkt uenig 1	2	3	4	Sterkt enig 5
Jeg liker å jobbe med Use Case					
Det er kjedelig å jobbe med Use Case.					
Jeg foretrekker andre metoder enn Use Case					
Hvis man har skjermbilde prototyp så er det ingen vits i Use Case.					
Veldig detaljerte use case fører til at gjennomføringen av prosjektet går bedre/raskere og med mindre feil.					
Jeg foretrekker mange små Use Case, fremfor ett stort et.					
Jeg tror det er en fordel at flere samarbeider om å skrive et Use Case.					

4.5 (*Kun ett kyss er tillatt pr spørsmål)*

	Ingen ganger	1 gang	2 eller flere ganger
Jeg har erfart at Use Case man har brukt mye tid på i begynnelsen har blitt irrelevante etter at utviklingen har startet			
Jeg har opplevd at Use Case har blitt så omfattende at jeg ikke har giddet å lese gjennom hele Use Caset.			
Jeg har opplevd at utviklere har programmert			

på Use Case som var utdaterte.			
Jeg har erfart at dersom en person har ansvaret for Use Casene, så blir Use Casene bedre vedlikeholdt og lettere å forholde seg til.			
Jeg har erfart at vi av og til fokuserer for mye på Use Casene, og glemmer litt andre deler av spesifikasjonen.			
Jeg har erfart at arbeidet med Use Casene har tatt så lang tid at det har hindret framgang i prosjektet			
Jeg har opplevd at Use Case beskrivelsene ikke er presise nok			
Jeg har erfart at Use Casene ikke bør inneholde detaljer om brukergrensesnittet			

4.6 Hvis du foretrekker andre metoder enn Use Case, hvilken metode foretrekker du?

☐ User stories
☐ Annet: _____

Begrunnelse:

4.7 Jeg syns Use Case ikke bør brukes til å*(Flere kryss tillatt)*

☐ Skrive brukerdokumentasjon
☐ Strukturere kravspesifikasjonen
☐ Programmere etter
☐ Lage opplærings kurs
☐ Estimering
☐ Lage test case
☐ Teste direkte etter Use Casene
☐ Systemdokumentasjon
☐ Annet_____

4.8 Use Case bør skrives av *(Flere kryss er tillatt dersom du mener de bør jobbe sammen)*

☐ En Use Case spesialist
☐ Utvikler
☐ Oppdragsgiver
☐ Andre: _____

4.9 Jeg syns Use Case er nyttig for å *(Flere kryss tillatt)*

☐ Estimere prosjektet
☐ Strukturere prosjektet
☐ Få frem gode krav
☐ Få en innføring til et system man ikke kjenner
☐ Lage opplæringskurs
☐ Skrive brukerdokumentasjon
☐ Kommunisere med oppdragsgiver

Del V: Personlige erfaringer
Denne delen tar for seg positive og negative erfaringer med Use Case generelt, utover det som kom fram i forrige del.

5.1 Kan du beskrive noen positive erfaringer med bruk av Use Case? *(Bruk gjerne stikkordsform)*

5.2 Kan du beskrive noen negative erfaringer med bruk av Use Case? *(Bruk gjerne stikkordsform)*

Del VI: Forbedringer av Use Case teknikken
Denne delen tar for seg endringer i Use Case teknikken, og forslag til endringer som burde vært gjort for å gjøre teknikken bedre.

6.1 Etter å ha snakket med flere som bruker Use Case har vi fått tilbakemeldinger om at det er ønskelig med forbedringer av teknikken. Kryss av for de alternativene du tror ville vært nyttig.

- ☐ En ordliste med termer som blir benyttet på en konsistent måte
- ☐ En standardisering av oppsett og innhold i Use Casene
- ☐ Et verktøy som gjør det enklere å få oversikt over relaterte Use Case. Eks. en Wiki, hvor man kan linke mellom skjermbilder, forretningsregler osv.
- ☐ Et verktøy som gjør det mulig å ekspandere stegene og/eller delene av et Use Case etter behov.
- ☐ Et verktøy som sier i fra om vesentlige endringer i Use Casene

6.2 Har du noen forslag til forbedringer?

6.3 Eventuelle andre kommentarer

Appendix C

Templates from the Companies

This Appendix contains the Use Case templates we got from the companies we interviewed. The templates are written in the original language.

Company A's Use Case template

Skrevet av:	**Dato:**	**Versjon:**
Godkjent av:		**Status:**

Nivå	
Use Case ID	
Use Case navn	
Use Case beskrivelse	
Endringer	
Aktører	
Trigger	
Pre-betingelser	
Post-betingelser	
Scenariooversikt	
Scenario 1	Versjon: Opprettet: Sist oppdatert: Åpne spørsmål: Kommentarer: Post betingelser: Sceanriosteg:

Company B's Use Case template

Use Case Navn

Use Case Nummer

Forfatter

Versjon Nummer

KVALITETSSIKRINGSLOGG

Dato	Vers	Navn

ENDRINGSLOGG

#	Dato	Vers	Navn	Disiplin	Årsak	Kommentar

USE CASE VIEW
<…>

MÅLSETNINGER
<…>

AVGRENSNINGER I FORHOLD TIL INNEVÆRENDE LEVERANSE
<…>

FAGSPØRSMÅL
<…>

DOKUMENTREFERANSER

Ref. nr.	Tittel	Link
[1]	Datagrunnlag	Datagrunnlag

AKTØRER:
<beskrivelse av aktører og hvilke innkanaler de skal bruke >

ARBEIDSFLYT

Startbetingelser
1)
2)

Sluttbetingelser
1)
2)

Normalt forløp
100 <seksjon 1 – inndata>:
1)
2)
200 <seksjon 2 – validering>:
1)
2)
300 <seksjon 3 – beregninger>:
1)
2)
400 <seksjon 4 - posteringer>:
1)
2)
500 <seksjon 5 – opprett/lagre/struktur>:
1)
2)
600 <seksjon 6 – rapporter>:
1)
2)
700 <seksjon 7 – tilbakemelding til aktør>:
1)
2)

Alternativt forløp <navn>
100 <seksjon 1>:
1)
2)
200 <seksjon 2>:
1)
2)
300 <seksjon 3>:
1)
2)

SPESIELLE KRAV

Saksbehandlerklient: Spesifikasjon av felter og prototype av skjermbilder

Se Datagrunnlag [1] for beskrivelse og betydning av feltene.

Felt

Felt	Kriterier	Format	Feilmelding	Editerbart

EKSEMPLER – TESTCASE
<legg inn eksempler på beregninger, hendelsesforløp og annet som kan være til hjelp for utvikling og enhetstesting/integrasjonstesting>

VEDLEGG – UX MODELL
<skjema>

VEDLEGG – ANDRE DIAGRAMMER FRA UC MODELL
<skjema>

Company C's Use Case template

Dato	Versjon	Ansvarlig	Godkjent av

1. KORT BESKRIVELSE

2. FLYT

Funksjonell flyt

2.1 Hovedflyt

Hovedflyten beskrevet i nummererte steg sammen med en tabell.

Felt	I/O	Obligatorisk	Kommentar

2.2 Alternativ flyt

De ulike alternative flytene beskrives som underseksjoner. Hver enkelt alternativ flyt beskrives på samme måte som hovedflyten.

3. RESULTAT

Beskrivelse av resultat fra det Use Caset som er beskrevet.

4. SPESIELLE KRAV

4.1 Krav til systemet

4.2 Krav til øvrige systemer

4.3 Andre krav

5. HISTORIKK

Dato	Versjon	Endret av	Beskrivelse

Company D's Use Case template

Dokument historikk

Dato	Versjon	Beskrivelse	Forfatter

1. Kort beskrivelse

Denne kan også inneholde skjermbilde

2. Scenarier

2.1 Basis scenario

2.2. Alternative scenarier

Disse skrives som egne underkapitler. De kan også inneholde figurer.

3. Spesielle krav

4. Starttilstand

5. Sluttilstand

Appendix D

Results from the Survey

This Appendix presents the results from the survey.

Del I:	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
1_1	Navn (valgfritt)																																							
1_2	Navn på bedrift																																							
1_3	Stilling	Leder	Systemutvikler	Systemutvikler	Senior konsulent	Testansvarlig	Senior Systemutvikler	Senior Prosjektleder	Prosjektleder	Senior prosjektleder	Bid manager	Senior system utvikler	Senior sjefskonsulent	Overingenior	Produktsjef	Systemutvikler	Sjefskonsulent	Systemarkitekt	systemsjef	systemutvikler	Senior systemkonsulent	Systemutvikler	Systemutvikler	Kvalitetsansvarlig		Konsulent	Utviklingsansvarlig	Senior systemarkitekt	Senior konsulent/rådgiver	Systemutvikler	Teamleder Java	Konsulent	Analyst	Konulent	Konsulent	Konsulent	Systemutvikler	Senior prosjektleder	Overingenior	
1_4	Antall år i bransjen	15	11	7	9	10	16	8	2	13	13	7_8	25	12	22	5	11	5	22	6	6	16	9	7	11	3	7	23	10	6	14	2	1	1	1	5	8	10	20	
1_5	**Hva har du jobbet med i prosjektsammenheng?**																																							
	Testing			1	1	1		1	1	1	1	1		1		1		1	1	1		1		1			1	1	1		1			1		1	1	1	1	**24**
	Programmering	1	1	1	1	1	1	1	1	1	1	1		1		1	1		1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	**33**
	Kravspesifisering	1		1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1		1		1			1	1	1	1	1	1	1	1	1	1	1	1	1	**32**
	Prosjektledelse		1		1	1	1	1	1	1	1						1	1	1	1		1		1	1		1	1	1		1					1	1	1	1	**23**
	Annet																																							
	Design						1									1														1						1				**4**
	Systemarkitekt													1																										**1**
	Salgsprosesser og annen koordineering ut mot kunder														1																									**1**
	Systembeskrivelse / dokumentasjon																	1																						**1**
	Standardisering																					1																		**1**
	Modellering/prototyping																			1																				**1**
	Testleder																							1																**1**
	Kravinnsamling																										1													**1**
	Kunderelasjoner																										1													**1**
	Driftsettelse																										1													**1**
	Kurs/opplæring - Usability																											1												**1**
	Tilbudsarbeid																												1											**1**
	Analyse og Design																																		1					**1**
	System og virksomhetsarkitektur	1																																						**1**
1_6	**Hvor mange prosjekter har du vært med på som har brukt Use Case?**																																							
	1 – 4					1	1	1	1	1	1				1	1			1			1	1			1	1			1		1	1	1	1	1	1		1	**21**
	5 – 9	1		1	1							1	1	1				1		1	1			1	1				1		1						1		**14**
	10 eller flere		1														1											1												**3**
1_7	**Hvordan har du vært involvert med Use Case?**																																							
	Har skrevet Use Casene	1	1	1	1	1	1		1	1		1	1	1	1	1	1	1	1	1		1					1	1	1	1	1	1	1	1	1	1	1	1	1	**31**
	Har skrevet testcase ut i fra Use Casene					1		1		1		1				1	1	1		1		1		1			1		1											**12**

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
	Har testet ut i fra Use Casene					1		1		1	1			1		1				1	1			1			1		1		1										**12**
	Har programmert ut i fra Use Casene	1	1	1	1	1	1		1		1	1		1		1	1		1	1	1		1			1				1	1	1					1	1	1	**23**
	Annet:																																							
	Design ut ifra Use Casene						1																																	**1**
	Utarbeidet spesifikasjon fra UC								1																															**1**
	Skjermbildedesign basert på Use Case													1																										**1**
	Laget detaljert design ut i fra Use Case																																			1				**1**
1_8	**Hva slags type Use Case bruker dere?**																																							
	Tekstlige beskrivelser								1						1				1							1		1		1						1			1	**8**
	UML diagram, dvs strek-menn og sirkler.						1														1		1								1				1	1			**6**
	Begge	1	1	1	1	1		1		1	1	1	1	1		1	1	1		1		1		1			1		1		1		1	1	1			1		**24**
	Kommentar: Men nesten alltid med figurer som illustrerer problemsliillingene, men uten å bruke fromaliserte metoder som UML etc.																											1												
1_9	**Hvilken utviklingsmetode bruker dere i bedriften?**																																							
	RUP	1			1			1	1	1	1	1	1		1					1	1	1	1	1					1				1				1			**17**
	XP																1									1			1		1								1	**5**
	Vannfallsmodellen												1					1										1				1								**4**
	Annet																																							
	Scrum	1	1				1	1		1				1			1																							**7**
	DSDB													1																										**1**
	Egen metode													1					1																					**2**
	Har en intensjon om å bruke Agile metoder….															1																								**1**
	UML (????)																	1																						**1**
	Forsøker på smidige metoder, men har en del RUP-arv.																										1													**1**
	Bedriftsspesifikk metode																																1		1	1				**3**
	Bedriftsspesifikk metode																																	1						**1**
	MSF Agile																																				1			**1**
	Scrum "light"/Agile (nå i det siste) vannfall og I	1																																						**1**
Del II:																																								
2_2	**Har dere en mal for hvordan Use Case skal skrives i bedriften?**																																							
	Ja	1						1		1			1		1			1		1					1								1		1	1		1	1	**13**
	Ja, men vi bruker den ikke											1							1											1										**3**
	Nei		1	1	1	1	1		1		1			1		1	1				1	1		1		1	1	1	1		1	1					1			**20**
	Vet ikke																						1											1						**2**
2_3	**Hvis JA på forrige spørsmål, hvilken mal?**																																							
	RUP	1						1		1			1		1										1					1										**7**
	Alistair Cockburn																																						1	**1**
	Vet ikke																																							**0**

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
	Annet																																							
	Egen prosedyre (UML basert)																	1																						1
	Egen																		1																					1
	Modifisert RUP																			1																	1			2
	Egen metode																																1		1	1				3
2_4	**Hva skriver dere Use Casene i?**																																							
	Microsoft Word	1		1	1	1		1	1	1		1	1	1	1		1		1	1				1		1	1	1		1			1		1	1		1	1	24
	Microsoft Visio	1					1				1	1		1		1		1		1		1							1		1	1	1	1	1	1				16
	Rational Rose	1										1		1							1		1		1																6
	Annet																																							
	Figurer laget i PPT. Bruker også mer og mer MindManager																											1												1
	Enterprise Architect																																				1			1
	Together		1	1																																				2
2_5	**Hvordan organiserer dere Use Case som er relatert til hverandre?**					Ubesvart																																		
	Vi bruker ikke noe spesielt verktøy for å organ Use Casene.			1	1		1	1	1			1		1	1			1	1	1	1		1	1		1	1	1	1		1	1	1	1	1	1	1		1	26
	På en Wiki									1						1														1										3
	Annet																																							
	Enterprise architect										1																														1
	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
	Filstruktur bygget på domene/leveranse												1																											1
	Visio																1																							1
	Word																																					1		1
	Sharepoint Prosjektrom	1																																						1
	Together		1																																					
2_6	**Hvordan organiserer dere Use Case og relaterte dokumenter?**					Ubesvart																																		
	Vi relaterer ikke Use Casene til andre dokumenter						1		1			1	1			1	1	1				1	1	1					1			1		1		1	1		1	16
	På en Wiki									1					1															1										3
	Annet														1																									1
	Interne ref. i dokumentene listet opp							1																																1
	Enterprise architect										1																														1
	Varierer veldig fra prosjekt til prosjekt													1																										1
	Hyperlinker i Use Case-dokumentene																		1																					1
	I egen tematisert/gruppert dokumentdatabase																			1																					1
	Kopi av diagrammet i et word-dokument																				1																				1
	Skriver inn referansen i dokumentet vi lager UC i																										1														1
	Visio, Word																														1										1
	Skriver inn ref til krav																																1								1
	Setter inn use casene i eget dokument med arkivreferanse																																		1						1

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
	Word		1																																			1		2
	Linker i tekstbeskrivelser	1																																						1
	I katalogstruktur			1																																				1
	Ved referanser i use casene, eller gjennom kravm				1																																			
2_7	**Hva blir Use Case brukt til i din bedrift?**																																							
	Å skrive brukerdokumentasjon		1			1							1		1		1	1	1	1	1		1	1								1		1				1		14
	Strukturere kravspesifikasjonen	1	1	1	1		1	1	1	1	1	1	1	1	1		1	1		1		1	1			1	1	1	1		1	1	1	1	1	1		1		29
	Programmere etter	1	1	1	1		1	1	1		1	1	1	1	1	1	1	1	1	1	1		1	1		1	1	1			1						1	1	1	27
	Å lage opplæringskurs												1				1			1								1												4
	Estimering	1	1		1		1	1		1	1		1	1			1		1	1				1		1	1		1	1	1	1		1		1		1	1	23
	Lage test case	1	1					1		1		1	1	1	1		1	1		1				1		1	1		1		1	1		1				1	1	20
	Teste direkte etter Use Casene							1									1									1														3
	Systemdokumentasjon		1	1									1	1			1	1			1		1	1		1		1	1	1				1				1		15
	Annet																																							
	Design						1																													1				2
	Avstemme med kunde/prototype												1																											1
	Kommunikasjon med andre bedrifter som bruker våre komponenter i sine produkter																					1																		1
2_8	**Vi bruker disse hjelpemidlene ved siden av Use Case**																																							
	Skjermbildeprototyp	1	1	1	1		1	1		1	1	1	1	1	1		1		1	1	1		1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	32
	Sekvensdiagram	1	1		1		1		1	1		1		1		1	1	1		1	1	1	1	1			1		1		1		1	1	1	1	1	1		25
	Aktivitetsdiagram	1			1		1									1		1		1									1		1			1	1		1	1	12
	Navigeringskart				1			1			1						1							1					1					1	1	1				9
	Bruker ikke hjelpemiddel					1																																		1
	Annet																																							
	Tradisjonelle brukes fortsatt																	1																						1
	Flytskjema																		1																					1
	Arbeidsflytdiagram (Visio)																																						1	1
	Use case diagrammer. Noen ganger generelle flytskjemaer i stedet for aktivitetsdiagrammer eller sekvensdiagrammer																																							
	Kommer an på hva som kommuniserer best i forh				1																																			
2_9	**Use casene blir først og fremst skrevet for**					Ubesvart																																		
	Oppdragsgiveren	1						1			1		1	1			1				1											1	1	1	1	1			1	13
	Utviklerne	1	1	1	1		1		1	1		1			1	1		1	1	1			1	1		1	1	1	1	1	1		1			1	1			24
	Testerne																												1											1
	Annet																																							
	Samarbeidspartner																					1																		1
	Funksjonelt team																																			1				1
	Hele prosjektgruppa, men først og fremst for å dokumentere krav																																				1			1
2_11	**Use Case beskrivelsene inneholder**					Ubesvart																																		
	Forretningsregler	1	1		1		1			1		1		1	1		1	1						1		1	1		1	1	1				1			1		18

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
	Aktør	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1		1	1	1	1	1	1	1	1	35
	Pre-betingelser	1		1	1			1	1	1	1		1	1	1	1	1		1	1		1					1		1		1		1		1	1		1	1	23
	Post-betingelser	1		1	1			1		1	1		1	1	1	1	1		1	1		1					1		1		1		1		1	1		1		21
	Endringslogg							1		1			1		1					1						1							1		1			1		9
	Hovedflyt	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1		1	1	1	1	1	1	35
	Alternativ flyt	1	1	1	1		1		1	1	1	1	1	1	1	1	1			1		1	1		1	1	1		1		1				1	1		1	1	26
	Andre felter																																							
	Krav									1																														1
	Regler									1																														1
	Diagrammer									1																														1
	Resultatet beskrives												1																											1
	Trigger																			1																				1
	Veldig avhengig av problemstillingen																											1												1
	Feilhåndtering (på overordnet niva)																																			1				1
	For dokumentorienterte aktiviteter: tilgjengelige dokumenttyper (maler)																																					1	1	1
	Triggere, Kort beskrivelse, åpne spørsmål				1																																			
2_12	**Våre Use Case inneholder antagelser om bruker-grensesnittet. Eks. "Trykk lagre knapp"**	Ubesvart																																						
	Sant		1	1						1		1	1		1		1	1		1	1		1			1	1	1		1				1		1		1	18
	Usant				1		1	1	1		1					1			1			1		1					1			1		1		1			13
	Vet ikke	1												1										1					1			1		1					6
2_13	**Hvor mange sider er en gjennomsnitts Use Case beskrivelse på?**	Ubesvart																																						
	1 til 5	1	1	1	1		1		1		1	1		1		1	1	1	1			1	1	1	1		1	1		1		1		1		1	1	1	1	26
	6 til 10							1		1			1		1					1	1																		6
	10 til 20																									1			1		1		1		1					5
	20 eller flere																																							
Del III:																																							
3_1	**Use Case alene fungerer bra som kommunikasjons-middel med oppdragsgiver**														Ubesvart																									
	1						1										1	1					1			1														5
	2			1	1	1		1	1	1				1								1										1					1	1	1	12
	3	1										1								1									1	1				1						6
	4										1		1						1		1			1	1		1				1		1			1				10
	5															1												1							1						3
	Jeg syns Use Case er en god teknikk å bruke til å kommunisere krav til oppdragsgiver når oppdragsgiver ikke er så kjent med IT-teknologi																																							
	1																1									1														2
	2			1	1		1							1																								1		5
	3	1							1	1										1									1			1								6

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
	4					1		1			1	1	1					1	1			1		1	1		1			1	1		1			1	1		1	17
	5															1					1		1					1							1	1				6
	Skjermbildeprototyp bør brukes i tillegg til Use Case fordi oppdragsgiver er visuell av seg og trenger et bilde av hvordan systemet skal se ut																																							
	1																																							0
	2																																							0
	3															1																								1
	4	1			1				1			1		1			1	1	1			1			1						1		1	1				1	1	15
	5			1		1	1	1		1	1		1							1	1		1	1		1	1	1	1		1			1	1	1	1			20
	Det ideelle er å bruke Use Case innad i prosjektet, og skjermbildeprototyp mot oppdragsgiver																																							
	1					1											1						1			1														4
	2	1						1		1	1		1	1		1			1		1			1	1					1				1				1		14
	3								1			1						1		1								1	1		1	1			1	1			1	11
	4			1	1		1															1					1						1				1			7
	5																																							0
	Jeg har fått tilbakemelding om at oppdragsgiver heller vil snakke om skjermbilder enn om Use Case																																							
	1																	1					1											1	1		1			5
	2																		1																	1				2
	3	1		1	1				1				1								1	1		1	1		1				1						1		12
	4					1	1	1			1	1		1							1								1	1	1							1	11
	5									1							1									1	1						1							5
	Jeg har fått tilbakemelding om at oppdragsgiver syns det er bortkastet å bruke Use Case																																							
	1				1	1	1						1			1		1			1		1		1			1	1		1			1	1	1	1			16
	2	1							1					1					1	1		1		1								1	1					1	1	11
	3			1				1			1	1					1									1				1										7
	4									1																														1
	5																										1													1
	Jeg har fått tilbakemelding om at oppdragsgiver syns det blir vanskelig å få oversikt over Use Casene																																							
	1					1										1		1			1		1		1			1									1			8
	2	1							1			1										1		1							1		1	1	1					9
	3			1	1			1									1		1	1									1	1		1							9
	4						1			1	1		1	1													1		1									1	1	9
	5																									1														1
Del IV																																								
4_1	**Jeg syns det er vanskelig å....**																																							

..Holde Use Casene oppdaterte gjennom prosjektet																																								
	1					1																1							1	1											4
	2	1											1					1																					1	4
	3		1				1								1	1			1								1								1				1				8
	4			1	1			1	1	1		1		1						1								1			1	1		1							12
	5										1						1					1	1	1		1								1	1		1	1		10
Total antall svar																																								38
..Vite hvilket nivå man skal legge seg på når man skriver Use Casene																																								
	1																											1												1
	2	1				1	1								1							1			1				1							1			1	9
	3		1					1				1	1										1				1				1		1					1		9
	4			1	1				1		1			1		1	1		1	1	1			1						1		1		1	1					15
	5									1							1								1												1			4
																																								38
...Holde styr på endringer i Use Casene																																								
	1																			1								1											2
	2	1				1							1		1													1		1								1		7
	3		1				1	1										1							1		1					1			1	1				9
	4			1	1				1	1		1		1		1			1	1				1							1		1	1					1	14
	5										1						1					1	1			1											1			6
																																								38
...Se sammenhengen mellom relaterte Use Case																																								
	1																					1							1											2
	2	1	1	1		1	1					1	1					1			1				1	1	1	1		1					1			1		16
	3							1	1	1				1	1	1			1	1			1	1							1	1	1	1			1			15
	4				1						1																									1			1	4
	5															1																							1
...Finne rett nivå på en Use Case beskrivelse som passer både utviklere, testere og oppdragsgiver																																								
	1																											1												1
	2	1				1																									1						1			4
	3		1	1					1				1		1									1	1								1							8
	4				1			1		1		1				1		1	1	1	1						1		1	1		1			1	1		1	1	17
	5						1				1			1			1					1	1			1								1						8
....Få kommunisert ut endringer i Use Case til utviklere og testere																																								
	1																											1	1											2
	2	1	1			1															1		1														1			6
	3			1			1	1		1			1	1		1		1				1		1			1					1	1		1	1				15
	4				1				1			1							1	1					1					1	1							1	1	10

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
		5									1						1									1								1						4
	...Prioritere Use Case																																							
		1						1					1									1							1											4
		2	1	1			1			1					1			1				1				1				1							1	1		11
		3				1					1	1				1			1	1				1				1		1	1	1		1	1			1	14
		4			1	1								1													1												4
		5							1								1						1			1									1						5
	...Skrive Use Case																																							
		1			1								1		1										1		1	1	1				1							8
		2	1	1		1		1		1	1				1					1		1								1	1			1	1	1	1		15
		3					1		1	1			1				1	1	1		1			1									1						1	11
		4												1																							1		2
		5																					1			1														2
4_2	Detaljnivået på Use Casene er passe							Ubesvart																																
		1				1																				1														2
		2								1	1	1				1					1										1						1			7
		3	1		1		1			1				1	1		1		1	1			1	1			1		1				1	1				1		16
		4		1				1										1				1			1						1	1			1			1	9
		5											1														1							1						3
	Tiden vi bruker på å skrive Use Case er passe							Ubesvart																																
		1				1	1																			1														3
		2							1	1		1		1		1						1	1				1											1	1	1	11
		3	1	1							1				1		1	1	1	1				1	1				1					1	1					13
		4			1			1																		1			1		1	1	1	1			1			9
		5											1																											1
	Tiden vi bruker på å lese Use Case er passe							Ubesvart																																
		1				1	1																			1								1						4
		2							1	1		1			1	1						1					1											1	1		9
		3	1	1	1												1	1	1	1	1		1	1				1	1							1				13
		4						1			1		1													1					1	1	1	1		1				1	10
		5											1																											1
	Tiden vi bruker på å oppdatere Use Case er passe							Ubesvart																																
		1				1																1				1														3
		2		1			1			1		1				1											1											1	1		8
		3	1		1						1			1	1		1	1	1	1	1		1	1	1			1	1	1	1									17
		4						1				1																					1	1		1	1		1	7
		5											1																											1
	Vi legger passe vekt på å holde Use Casene oppdaterte	Ubesvart																																						
		1				1																1															1			3
		2					1	1		1	1		1			1	1		1									1	1		1	1				1		1		14
		3			1	1								1			1			1	1		1	1	1	1														10

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
	4	1									1								1										1			1	1			1			1	8
	5												1																											1
4_3	Jeg syns Use Case passer i følgende typer prosjekter																																							
	...Nyutviklingsprosjekter																																							
	1																									1														1
	2																													1										1
	3				1												1																							2
	4	1		1					1	1	1			1					1					1	1						1	1					1		12
	5		1			1	1	1				1	1		1	1		1		1	1	1	1				1	1	1				1	1	1	1	1		1	22
	...Forvaltningsprosjekter																																							
	1																									1								1						2
	2				1					1			1					1											1		1					1	1		1	9
	3	1	1	1				1			1		1		1		1							1										1			1			11
	4						1		1			1				1			1								1			1			1	1						9
	5					1														1	1	1	1		1															6
	...Internett-portal																																							
	1																									1												1		2
	2																1																	1						2
	3	1		1					1					1								1			1				1			1				1			9
	4				1	1		1		1	1	1			1	1		1	1					1					1		1	1						1	15
	5						1						1							1	1		1			1									1	1				8
	...I prosjekter hvor man er nær oppdragsgiver																																							
	1																									1														1
	2																1																						1	2
	3																												1	1							1			3
	4	1		1	1	1	1	1	1	1	1	1		1	1			1	1		1			1	1		1				1	1		1			1		22
	5												1			1				1		1	1					1					1		1	1				9
4_4	Jeg liker å jobbe med Use Case																																							
	1																									1														1
	2										1																			1									1	3
	3			1			1		1				1		1		1		1				1											1		1	1			11
	4	1	1		1			1		1		1		1		1		1			1	1		1	1		1		1		1	1	1		1				1	20
	5					1														1								1												3
	Det er kjedelig å jobbe med Use Case																																							
	1							1							1	1		1		1		1			1			1										1			9
	2	1	1						1	1		1		1					1		1			1		1		1		1		1						1	1	15
	3			1	1		1						1			1							1						1		1		1		1					10
	4					1					1																								1					3
	5																									1														1
	Jeg foretrekker andre metoder enn Use Case																																							
	1												1		1	1																						1			4

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
	2	1	1		1	1	1		1					1				1	1	1	1	1	1		1			1	1		1	1						1		19
	3			1				1		1	1	1					1							1			1						1	1	1				11
	4																													1						1				2
	5																									1												1		2
	Hvis man har skjermbildeprototyp så er det ingen vits i Use Case																																							
	1					1		1		1		1	1	1	1	1	1	1	1	1	1	1	1		1			1	1	1					1			1	1	22
	2	1	1		1		1		1		1																1			1	1	1			1				11
	3			1																				1										1						3
	4																																				1			1
	5																									1														1
	Veldig detaljerte Use Case fører til at gjennomføringen av prosjektet går bedre/raskere og med mindre feil																																							
	1							1									1	1								1														4
	2					1	1			1		1		1																1								1	1	8
	3	1		1	1						1					1			1					1			1	1	1			1	1		1				13
	4		1						1						1					1	1	1			1						1			1		1		1		11
	5												1										1																	2
	Jeg foretrekker mange små Use Case, fremfor ett stort et																																							
	1																																							0
	2				1																													1					2
	3	1		1			1	1							1		1			1					1				1	1					1				11
	4		1			1			1					1					1		1			1		1					1	1	1			1	1	1	14
	5									1	1	1	1			1		1				1	1			1		1											1	11
	Jeg tror det er en fordel at flere samarbeider om å skrive et Use Case																																							
	1																																							0
	2																																				1			1
	3																																1							1
	4	1	1	1	1	1	1	1			1				1		1	1	1			1		1	1			1			1	1		1	1	1			1	22
	5								1	1		1	1	1		1				1	1		1			1	1		1	1								1		14
4_5	Jeg har erfart at Use Case man har brukt mye tid på i begynnelsen har blitt irrelevante etter at utviklingen har startet														Ubesvart																									
	Ingen ganger					1		1					1					1			1	1							1			1	1		1	1				11
	1 gang								1	1										1								1										1		5
	2 eller flere ganger		1	1	1		1				1	1		1		1	1		1				1	1	1	1	1			1	1			1			1		1	20
	Jeg har opplevd at Use Case har blitt så omfattende at jeg ikke har giddet å lese gjennom hele Use Caset																																							
	Ingen ganger		1		1		1		1		1		1					1		1	1	1	1						1	1		1			1	1	1			17

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
	1 gang					1																		1	1			1					1							5
	2 eller flere ganger			1			1		1		1		1		1	1			1							1	1				1			1				1	1	14
	Jeg har opplevd at utviklere har programmert på Use Case som var utdaterte																																							
	Ingen ganger	1	1	1		1		1		1	1	1	1				1	1		1	1	1	1	1	1		1	1	1	1			1	1		1	1	1	1	26
	1 gang																															1								1
	2 eller flere ganger					1		1		1						1	1			1						1					1										8
	Jeg har erfart at dersom en person har ansvaret for Use Casene, så blir Use Casene bedre vedlikeholdt og lettere å forholde seg til.																																							
	Ingen ganger	1	1	1							1	1		1		1	1	1	1			1			1				1	1		1		1			1			17
	1 gang					1			1																	1														3
	2 eller flere ganger						1	1		1			1							1	1		1	1			1	1			1		1		1	1		1	1	16
	Jeg har erfart at vi av og til fokuserer for mye på Use Casene, og glemmer litt andre deler av spesifikasjonen																																							
	Ingen ganger	1	1	1	1							1				1		1	1	1	1	1			1		1	1	1	1	1	1	1	1	1	1	1		1	24
	1 gang						1	1			1												1																	4
	2 eller flere ganger								1	1			1	1			1							1		1													1		8
	Jeg har erfart at arbeidet med Use Casene har tatt så lang tid at det har hindret framgang i prosjektet																																							
	Ingen ganger	1	1	1		1	1	1			1	1		1		1		1	1		1	1			1	1	1	1	1		1	1	1	1		1	1		1	25
	1 gang									1	1			1										1											1			1		6
	2 eller flere ganger					1											1			1			1							1										5
	Jeg har opplevd at Use Case beskrivelsene ikke er presise nok																																							
	Ingen ganger									1												1												1					1	4
	1 gang																								1							1	1				1			4
	2 eller flere ganger	1	1	1	1	1	1	1	1		1	1	1	1		1	1	1	1	1	1		1	1		1	1	1	1	1	1				1	1		1		28
	Jeg har erfart at Use Casene ikke bør inneholde detaljer om brukergrensesnittet.																																							
	Ingen ganger	1	1	1									1							1			1	1			1		1	1		1	1	1		1	1			15
	1 gang					1			1																														1	3
	2 eller flere ganger						1	1		1	1	1	1			1	1	1	1		1				1	1		1			1				1			1		17
4_6	**Hvis du foretrekker andre metoder enn Use Case, hvilken metode foretrekker du?**	Ubesvart																																						
	User stories	1			1			1		1	1						1									1												1		8
	Annet																																							
	Skjermbildeprototyp og testbeskrivelser									1																														1
	Prototyp				1						1																													2
	Muntlig kommunikasjon																1																							1

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
	Kode																1																								1
	Aktivitetsdiagram, sekvensdiagram, flytskjema																	1																							1
	Prototyping (i tillegg til UC'ses)																										1													1
	Detaljert kravspesifikasjon evt. I tillegg til Use Case																																			1				1
	Kombinasjon av skjermbilder og use case er et must. Skjermbilder viser ikke funksjonalitet																																					1		1
	Arbeidsflytdiagrammer																																						1	1
	Skjermbilder med beskrivelse			1																																				1
4_7	**Jeg syns Use Case ikke bør brukes til å**					Ubesvart																																		
	Skrive brukerdokumentasjon			1	1		1			1	1	1		1								1				1											1		1	11
	Strukturere kravspesifikasjonen							1													1									1										3
	Programmere etter									1						1		1			1				1										1						6
	Lage opplæringskurs				1		1		1	1	1			1				1				1				1										1	1			11
	Estimering															1	1								1	1				1				1	1		1			8
	Lage testcase																				1				1												1			3
	Teste direkte etter Use Casene								1	1				1	1			1			1													1	1	1	1			10
	Systemdokumentasjon							1		1	1	1					1				1					1					1									8
	Annet																																							
	Som utgangspkt for detlajert design																																			1				1
4_8	**Use Case bør skrives av**																																							
	En Use Case spesialist	1	1	1	1			1		1	1	1	1	1	1	1				1	1	1			1	1	1			1	1	1	1							22
	Utvikler	1	1	1			1		1		1	1			1	1	1	1	1	1	1	1	1	1	1			1	1			1	1	1	1	1		1		1	26
	Oppdragsgiver			1		1		1	1	1	1	1		1	1	1			1	1	1	1	1	1	1		1	1		1					1		1	1		1	24
	Den som spesifiserer(trenger ikke være spesialist)					1																																		1
	Alle som skal beskrive funksjonelle krav												1																												1
	Noe avhengig av prosjektet														1									1																2
	Tester/testleder																																							0
	Medarbeidere som har mye kundekontakt samt kvailtetsansv.																			1																				1
	Å skrive usecase bør være en felles prosess der flere deltar																												1											1
	Funksjonelle og tekniske arkitekter																																		1					1
	Funksjonelt team /funksjonell arkitekt																																			1				1
	Kravansvarlig el.l																																					1		1
	Andre																																							
4_9	**Jeg syns Use Case er nyttig for å**																																							
	Estimere prosjekt	1	1		1		1	1	1	1	1			1	1				1	1		1	1				1			1		1						1	18
	Strukturere prosjektet	1	1	1	1		1			1	1	1		1	1		1		1	1	1	1	1	1			1	1		1		1	1							21
	Få frem gode krav	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1		1		1	1			1		1	1	1	1	1		1	31
	Få en innføring til et system man ikke kjen		1		1	1	1			1	1	1		1	1	1	1	1	1	1	1	1	1	1			1	1			1		1	1			1		1	25
	Lage opplæringskurs					1													1				1																	3
	Skrive brukerdokumentasjon		1			1												1	1				1					1												6
	Kommunisere med oppdragsgiver	1		1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		1			1	1		1	1	1	1	1	1	1	32
	Se 2.7																												1											1

	Deltager nr	35	36	37	38	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
Del VI:																																								
6_1	**Etter å ha snakket med flere som bruker Use Case har vi fått tilbakemeldinger om at det er ønskelig med forbedringer av teknikken. Kryss av for de alternativene du tror ville vært nyttig.**	Ubesvart																																					
	En ordliste med termer som blir benyttet på konsistent måte	1	1	1		1		1	1			1	1		1	1	1	1	1			1		1	1	1						1	1	1	1				1	**22**
	En standardisering av oppsett og innhc	1		1	1		1			1	1	1		1		1		1	1	1	1		1				1		1		1	1	1	1	1	1		1	1	**24**
	Et verktøy som gjør det enklere å få oversi relaterte Use Case.	1	1	1		1	1			1		1		1		1	1	1	1	1	1			1	1	1		1	1	1	1		1	1	1		1	1	1	**27**
	Et verktøy som gjør det mulig å ekspander og/ekker delene av et Use Case etter behov	1			1			1				1					1			1			1							1	1						1			**10**
	Et verktøy som sier ifra om vesentlige endringer i Use Case				1			1				1	1			1	1			1		1	1	1	1	1			1		1				1					**15**

Glossary

BI	Business Intelligence, an application that gather and structure information and makes it easier to make the right decisions.
Client	The employees in the company who pays for the system.
CRC cards	A technique that documents collaborative design decisions on index cards, (Beck and Cunningham, 1989).
CRM	Customer Relationship Management, a database with information about the client.
IDI	Department of Computer and Information Science. See more on `www.idi.ntnu.no`
IEEE	The Institute of Electrical and Electronics Engineers, the world's leading professional association for the advancement of technology. See more on `www.ieee.org`
NTNU	Norwegian University of Science and Technology. See more on `www.ntnu.no`
SRS	Software Requirements Specification
TDD	Test Driven Development, a development technique. See more on `www.testdriven.com`
UML	Unified Modeling Language, (Fowler, 2003).

Use Case description	A Use Case written in a textual form.
Use case diagram	A graphical Use Case with stick-men and ellipses.
User	The end-user of the system.
Wiki	A program that allows users to collaborate in writing the content of a Web site.
XP	eXtreme Programming, a development methodology. See more on `www.extremeprogramming.org`
RUP	Rational Unified Process, a development methodology.

Bibliography

Anda, B., Sjøberg, D., and Jørgensen, M.: 2001, *Quality and Understandability of Use Case Models*, Springer-Verlag

Beck, K.: 1999, *Extreme Programming Explained: Embrace change*, Addison-Wesley

Beck, K. and Cunningham, W.: 1989, *A Laboratory For Teaching Object-Oriented Thinking*, ACM Press

Biddle, R., Noble, J., and Tempero, E.: 2001, *Role-play and Use Case Cards for Requirements Review*, Proceedings of the Twelfth Australasian Conference on Information Systems

Bobkowska, A.: 2005, *A Methodology of Visual Modeling Language Evaluation*, Springer-Verlag Berlin Hedelberg 2005

Cockburn, A.: 1997, *Structuring Use Cases with goals*, Journal of Object-Oriented Programming, SIGS Publications, Sep-Oct 1997 and Nov-Dec 1997.

Cockburn, A.: 2001, *Writing effective use cases*, Addison-Wesley

Cohn, M.: 2004a, *Advantages of User Stories for Requirements*, Prentice-Hall

Cohn, M.: 2004b, *User Stories Applied: For Agile Software Development*, Addison-Wesley

Constantine, L. and Lockwood, L.: 1999, *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*, ACM Press

Cornford, T. and Smithson, S.: 2006, *Project research in Information Systems: A Student's Guide*, Palgrave Macmillan

Cox, K.: 2000, *Cognitive Dimensions of Use Cases - feedback from a student questionnaire*, Proceedings of the Twelfth Annual Meeting of the Psychology of programming

Interest Group, Corigliano calabro, Cosenza, Italy, pp. 99-121.

Firesmith, D. G.: 1995, *Use Cases: the Pros and Cons*, ROAD, Vol.2,No2,pp2-6

Fowler, M.: 2003, *UML distilled third edition. A brief guide to the standard object modeling language*, Addison-Wesley

Goguen, J. A. and Linde, C.: 1993, *Techniques for Requirements Elicitation*, Software Requirements Engineering, 2nd. Ed., IEEE CS Press, 1997, pp 152-164.

IEEE: 1990, *IEEE Standard Glossary of Software Engineering Terminology*, Institute of Electrical and Electronics Engineers

IEEE: 1994, *IEEE Recommended Practice for Software Requirements Specifications*, Institute of Electrical and Electronics Engineers

Jacobson, I.: 2003, *Use Cases: Yesterday, Today, and Tomorrow*, The Rational Edge: e-zine for the Rational Community

Jacobson, I., Christerson, M., Jonsson, P., and Övergaard, G.: 1992, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley

Jenkins, N.: 2005, *How to Make Software, An introduction to software development*, Creative Commons

Kruchten, P.: 2003, *The Rational Unified Process: An Introduction*, Addison Wesley Professional

Kvale, S.: 1996, *InterViews: An introduction to Qualitative research interviewing*, Sage publications

Larman, C.: 1997, *Applying UML and patterns*, Prentice-Hall PTR

Larman, C.: 2004, *Agile and Iterative Development. A manager's guide*, Addison-Wesley

Lilly, S.: 2001, *How to Avoid Use-Case Pitfalls*, Software Development Magazine

L.Mordal, T.: 1989, *Som man spør får man svar*, S.Hammerstads Boktrykkeri

Nawrocki, J. and Olek, L.: 2005, *UC Workbench - A Tool for Writing Use Cases and Generating Mockups*, Springer-Verlag Berlin Heidelberg

Schwaber, K.: 1995, *The SCRUM development Process*

Sinnig, D., Rioux, F., and Chalin, P.: 2005, *Use Cases in Practice: A survey*, `http://www.dsinnig.com/pdfs/CUSEC05_Sinnig.pdf`

Stalheim, G. and Kjeøy, M.: 2006, *Use Cases as a tool for communication: An empirical study of the understandability of Use Cases*

Vliet, H. V.: 2000, *Software Engineering, Principles and Practice*, Wiley

Wedde, K. J.: 2000, *SPIQ: TKL - 7 ledelsesteknikker*, http://www.geomatikk.no/spiq/pulikasjoner/teknikknotater/TKL-nye.pdf

Wells, D.: 2006, *Extreme Programming: A gentle introduction*, http://www.extremeprogramming.org/map/project.html

Wiegers, K. E.: 1999, *Software Requirements*, Microsoft Press

Wirfs-Brock, R. and McKean, A.: 2001, *The Art of Writing Use Cases*, OOPSLA Conference, 2001.

Wirfs-Brock, R. J.: 1993, *Designing Scenarios: Making the Case for a Use Case Framework*, Smalltalk report

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., and Wesslén, A.: 2000, *Experimentation in Software Engineering An introduction*, Kluwer Academic Publishers