

ABSTRACT

Agile methods have the recent years gained ground within the software industry by being lean and flexible, and one of the companies having adopted agile methods is the fast-growing Swedish developer of mobile applications Tactel AB. The company was interested in enhanced knowledge within the agile area, specifically about Scrum, and with one of its development teams – Mobical – in focus, Tactel wanted to analyze what is described by the research question:

How and where can Scrum be lifted from an internal management tool to get the customers more involved, and where is customer value created?

For answering the research question, four propositions were identified investigating ways for a customer to be involved in the agile process, underlying factors that gain customer value, how Mobical's competitors work, and how Mobical can benefit from the research.

The research was approached qualitatively by a multiple case study where flexible semi-structured interviews were the main sources of information. As a foundation for the data collection and analysis, theoretical models for customer classification and customer value analysis were identified and evolved from the literature. These models are referred to as the *Scrum Customer-classification System (SCS)* and the *Critical Success Factors (CSFs)*.

Two cases besides Mobical were selected to investigate; an outsourced Sony Ericsson project and a project at TAT, The Astonishing Tribe, AB. By the case studies the SCS and CSF models were further evolved, the studied projects benchmarked, and the propositions investigated.

The benchmark showed differences and similarities between the projects, and identified several areas for Mobical to improve; among others co-locating of all project-members, definition of *done*, and the team-members motivation. The research points out that customer value is created by factors identified by the CSFs, and that the classification levels in the SCS can describe how to get customers more involved. Though, the customer must experience that the total benefits with a deeper engagement are greater than the total sacrifices – only then is customer value created.

Keywords

Agile, Scrum, customer, value, classification, process, analysis

SAMMANFATTNING – SUMMARY IN SWEDISH

Agila metoder har de senaste åren genom sin flexibilitet och sitt *lean thinking* vunnit terräng inom mjukvarubranschen. Ett av de företag som börjat implementera agila metoder är den snabbväxande svenska tillverkaren av mobila applikationer Tactel AB. Företaget var intresserat av att fördjupa sina kunskaper inom det agila området, framförallt Scrum, och genom att fokusera på ett av sina utvecklingsteam – Mobical – ville Tactel analysera det som beskrivs av undersökningens huvudfrågeställning:

Hur och var kan Scrum lyftas från ett internt projektverktyg till att involvera kunderna djupare, och var skapas kundvärde?

För att besvara huvudfrågeställningen togs fyra delfrågor fram. Dessa undersökte på vilka vis en kund kan engageras djupare i den agila processen, underliggande faktorer som skapar kundvärde, hur Mobicals konkurrenter arbetar och hur Mobical kan dra nytta av studiens resultat.

Arbetet genomfördes som en kvalitativ flerfallsstudie där flexibla semistrukturerade intervjuer tjänat som huvudsakliga informationskällor. Genom litteraturstudier togs teoretiska modeller för klassificering av kunder och kundvärdesanalys fram. Modellerna som i studien benämns *Scrum Customer-classification System (SCS)* och *Critical Success Factors (CSFs)*, har legat till grund för datainsamling och analys

Jämte Mobical valdes två fall att studera ut; ett outsourcat Sony Ericsson-projekt och ett projekt på TAT, The Astonishing Tribe, AB. Fallstudierna ledde till en ytterligare utveckling av SCS- och CSF-modellerna, en benchmark av projekten samt att studiens delfrågor undersöktes.

Genom benchmarken påvisades likheter och skillnader i projekten, och flera områden för Mobical att utveckla identifierades; exempelvis samlokering av alla projektmedlemmar, definition av *done* och team-medlemmarnas motivation. Undersökningen visar att kundvärde skapas genom faktorer identifierade av CSF, och att de olika klassificeringarna i SCS kan användas för att beskriva hur en kund engageras djupare. Dock måste kunden uppleva att de totala vinsterna med ett djupare engagemang är större än de totala uppoffringarna – endast då skapas kundvärde.

PREFACE

This master's thesis is the concluding part of my Master of Science degree in Electronic Engineering at the Faculty of Engineering, Lund University. The research is benchmarking three projects working with agile methods, with focus on identifying where customer value is created. The project has been performed during fall 2008 at the department of Industrial Management and Logistics, Lund University in cooperation with Tactel AB.

The recent years I have worked with project management in various contexts, and the last two at Tactel. Within the company I have gotten familiar with agile methods, and was inspired to gain deeper knowledge within the up-coming area by this research.

I would like to thank Tactel AB with my supervisor Jonas Falk for great support and valuable input and my supervisor at the department of Industrial Management and Logistics Bertil I Nilsson for interesting discussions and invaluable input. I also would like to thank TAT, The Astonishing Tribe, AB for giving me access to and helping me get in contact with the projects to study, and I am very grateful for all of you taking part in the study, giving me insights in how you do Scrum and extensive material for the study.

Lund and Malmö, December 2008

Henrik Svenbrant

TABLE OF CONTENT

ABSTRACT	I
SAMMANFATTNING – SUMMARY IN SWEDISH	III
PREFACE	V
TABLE OF CONTENT	VII
1 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 AGILE SOFTWARE DEVELOPMENT	1
1.3 COMPANY PRESENTATION.....	3
1.4 PROBLEM DEFINITION AND PURPOSE	3
1.5 FOCUS AND DELIMITATIONS.....	4
1.6 DISPOSITION.....	5
1.6.1 Reading instructions	6
2 METHODOLOGY	7
2.1 RESEARCH PURPOSE	7
2.1.1 Research purpose – approach and considerations.....	7
2.2 RESEARCH STRATEGY.....	8
2.2.1 Research methods	8
2.2.2 Fix and Flexible methodologies	8
2.2.3 Quantitative and Qualitative research	9
2.2.4 Research strategy – approach and considerations.....	9
2.3 CASE STUDIES.....	10
2.3.1 Research design	10
2.3.2 Case study designs.....	10
2.3.3 Case studies – approach and considerations.....	11
2.4 DATA COLLECTION	13
2.4.1 Interviews	14
2.4.2 Observations.....	14
2.4.3 Written documentation.....	14
2.4.4 Data collection – approach and considerations	14

TABLE OF CONTENT

2.5	PRACTICAL APPROACH	16
2.5.1	Startup and planning and Problem definition	16
2.5.2	Literature study.....	17
2.5.3	Research design	17
2.5.4	Data collection and Analysis	18
2.5.5	Discussion and conclusion	18
2.6	SOURCES OF CRITICISM	18
2.6.1	Research quality.....	18
2.6.2	Quality – approach and considerations	19
3	THEORY	21
3.1	AGILE METHODOLOGY	21
3.1.1	Basics of agile methodology.....	22
3.1.2	Benefits and limitations	22
3.2	SCRUM	23
3.2.1	Scrum overview.....	24
3.2.2	Three Roles	25
3.2.3	Flow.....	25
3.2.4	Scaling	27
3.3	THE CUSTOMER AND SCRUM.....	28
3.3.1	Direct participant customers	29
3.3.2	Indirect participant customers.....	30
3.4	CUSTOMER VALUE.....	32
3.4.1	Value components models	32
3.4.2	Benefits/cost ratio models.....	33
3.4.3	Means-ends models.....	33
3.5	AGILE DRIVERS OF CUSTOMER VALUE	33
3.5.1	Critical Success Factors	34
3.5.2	CSFs – Reflection	35
3.5.3	The Drivers	40
3.5.4	The Drivers and SCS	42

TABLE OF CONTENT

4	THE CASES	45
4.1	DESIGNING THE CASE STUDY QUESTIONS	45
4.1.1	Background	45
4.1.2	Customer characteristics	46
4.1.3	Drivers of customer value.....	46
4.2	PERFORMING THE STUDIES	47
4.2.1	Criteria for the cases.....	47
4.2.2	Selecting the case candidates.....	47
4.2.3	Field procedures	49
4.3	THE CASES	50
4.3.1	Case A – Mobical.....	50
4.3.2	Case B – Audio Control	53
4.3.3	Case C – Cascades and Kastor.....	55
5	ANALYSIS.....	59
5.1	CASE A – MOBICAL	59
5.1.1	Case A – Model reinforcement 1	59
5.1.2	Customer classification	61
5.1.3	CSFs and Drivers	62
5.2	CASE B – AUDIO CONTROL.....	69
5.2.1	Case B – Model reinforcement 2	69
5.2.2	Customer classification	69
5.2.3	CSFs and Drivers	70
5.3	CASE C – CASCADES AND KASTOR	77
5.3.1	Case C – Model reinforcement 3	77
5.3.2	Customer classification	77
5.3.3	CSFs and Drivers	79
5.4	REINFORCED MODELS.....	86
5.4.1	Scrum Customer-classification System	86
5.4.2	Critical Success Factors	86
5.5	BENCHMARK	88
5.5.1	Project backgrounds and customers	88
5.5.2	Agile implementation	90
5.5.3	Benchmark summary	98

TABLE OF CONTENT

6	DISCUSSION AND CONCLUSIONS	99
6.1	THE STUDY PROPOSITIONS.....	99
6.1.1	Drivers of customer value	99
6.1.2	Customer classification	100
6.1.3	Benchmark	100
6.1.4	Application on Mobical	101
6.1.5	Success of the proposition study	103
6.2	THE MAIN QUESTION	104
6.3	FINAL CONCLUSIONS	105
6.3.1	Generalizations and limitations	105
6.3.2	Areas for further research	107
7	REFERENCES	109
	APPENDIX CASE STUDY DATA	111
	INVESTIGATION PROTOCOLS.....	111
	COLLECTED DATA.....	117
	Case A – Mobical.....	117
	Case B – Audio Control.....	129
	Case C – Cascades and Kastor	139

1 INTRODUCTION

This chapter presents the background, the studied company and the overall purposes and goals for this research. It also outlines focus and delimitations, and the disposition of the report.

1.1 BACKGROUND

Until today, in a global perspective, most software development has been carried out with traditional methodologies, with plans and requirements sometimes aiming years ahead. These methodologies are often not the most advantageous ones because of the rapid change of technique and customer wants in the software industry. Alternative, *agile*, development methods have been evolved around the world, and today it is more and more common in the industry to adopt those software development methods.

1.2 AGILE SOFTWARE DEVELOPMENT

In the traditional, plan-driven, ways of software development, basically requirements and deadlines are agreed at project start. The process of changing the requirements when a project is running is often rigorous and involving senior management and change control boards. The traditional approach claims that problems can be fully specified and that predictable solutions exist for every problem (1 p. 834).

This is in fact not the case in many software development projects, and as a reaction to the plan- and document-driven methodologies, software methodologists started to apply *lean* production philosophies when building software. Lean development, rooted in the Toyota Production System from the 1950s, focuses on eliminating waste (i.e. activities that do not add value for the customer), achieving quality first time and problem solving (1 p. 836).

Applying lean principles to software development has since the mid-1990s lead to the use of the term *agile*, and in 2001 a group of experienced software development methodologists defined their experience-based guidelines in the

INTRODUCTION

Manifesto for Agile Software Development (2). The manifesto states that agile development should focus on four core values:

- Individuals and interactions
- Working software
- Customer collaboration
- Responding to change

Several agile methods are in use around the world, more or less adapted to organizations' specific environments and needs, and the main methods are (1 p. 835):

- Extreme programming (XP)
- Scrum
- Lean software development
- Feature-driven development
- Crystal methodologies
- Dynamic software development method (DSDM)

Some of the basic techniques and characteristics common for the methods are as follows (3 pp. 21-22):

- *Small releases.* Work is divided into small packages and releases are usually delivered in one to three months.
- *Iterative and incremental development.* Plans, requirements, design, code and tests are evolved incrementally through multiple iterations.
- *Co-location.* Team members are collocated to facilitate face-to-face-communication.
- *Release plan/feature backlog.* Desired features are defined at a high level and prioritized by customers in a release plan or feature backlog.
- *Iteration plan/task backlog.* Lower-level features are defined and prioritized at the start of each iteration.
- *Self-organizing teams.* Team members self-organize without top-down management control.
- *Tracking.* Features and tasks are tracked within an iteration. They count as complete only when 100 percent done.
- *Simple, lean and adaptable.* All aspects of work, including processes, are kept simple, lean (low on waste), and adaptable.

INTRODUCTION

1.3 COMPANY PRESENTATION

Tactel AB is a developer of mobile applications, providing solutions and consulting services to many of the world's major operators and handset vendors (4). The company has had a large growth the last ten years, and employs about 350 people in Sweden, U.S.A. and Ukraine (October 2008). The company is private held with headquarters in Malmo, Sweden, and the major offices are located in Malmo and Lund – both in the Oresund region in the southern of Sweden. The company is divided in subunits, each managing several development projects. This study is mainly focusing on the Mobical development team located at the Malmo office.

Mobical is one of Tactel's own products, and is a service for synchronizing and backing up data on mobile phones. Main customers are network operators and service providers worldwide, offering the service to their subscribers. The core product – the synch and backup service – is adapted according to each customer's requests. The team developing Mobical consists of about ten people, and has been working according to Scrum methodologies since spring 2007.

Tactel has adopted agile methods in some projects, of which Mobical is one, and considers carrying on in more projects. The company experiences that the agile methods suit the needs of internal management well; that is to say this far the methods have mostly served as an internal process, with the result that work is done more efficiently and with less stress on the employees. The efficiency, i.e. shorter lead-times and improved quality, creates competitive advantages and increases value for the customers.

1.4 PROBLEM DEFINITION AND PURPOSE

One of the fundamental principles of agile methods, stated in numerous literary sources, is that the customer gets involved and takes part in the development process, and continuously is able to monitor and influence the progress. In the Mobical projects this is mostly done through the Product Owner, who acts as a customer proxy, prioritizing features and tasks to be developed. This means that the customers are not directly involved in the development process.

Tactel wants, with Mobical in focus, to achieve an increased knowledge of success factors that agile methods benefit in order to gain overall business value. The company wants to analyze how and in which cases to lift Scrum

INTRODUCTION

from an internal management tool to a level where the customers are more involved, aiming to improve the perceived customer value – and as a consequence the business value. This is done by answering the following question – the research question:

How and where can Scrum be lifted from an internal management tool to get the customers more involved, and where is customer value created?

In the first part of the question *how* means which factors that should be considered and *where* means for which types of customers and projects.

To define the research area and provide answers to the research question, the following sub-questions – propositions – are investigated:

Which drivers of customer value can be identified in agile practices?

How can customers be classified in matter of participation in an agile project?

Two additional propositions are also investigated to relate the research to Tactel and Mobical:

How well are the drivers and customer types considered by companies in Tactel's business context?

How can Mobical benefit from the findings?

The definition of the research area is used specifically when building the theoretical framework, and the collection and analysis of data is, as later described, designed to provide answers to the propositions and the research question.

1.5 FOCUS AND DELIMITATIONS

This study analyzes customer relations in agile projects in a general perspective and how Tactel AB can improve its use of agile methods. The study refers to the Mobical project, which is benchmarked with other projects in the same business context. The focus of the study is on Scrum in particular, but also refers to agile methods in general.

INTRODUCTION

1.6 *DISPOSITION*

This report is divided into the following sections:

Chapter 1 INTRODUCTION

An introduction to the research area is given. This section contains background, company presentation, problem definition, purpose and delimitations.

Chapter 2 METHODOLOGY

Various methodological approaches are presented, and which are selected in this research discussed. Also sources of criticism are identified.

Chapter 3 THEORY

The theoretical framework for this research is presented and discussed. First general agile theories are presented, followed by explanations of Scrum and its benefits and theories about customer value. This section also presents analysis models for customer classification and process analysis in agile projects.

Chapter 4 THE CASES

Selection of the cases to study, the design of questions for the study and how to approach the companies are described. Finally the three cases in the study are presented.

Chapter 5 ANALYSIS

Collected data from the case study is analyzed on the basis of the theories presented in chapter 3. First the cases are analyzed individually, followed by a cross-case analysis. The chapter also contains an evaluation of the analysis models.

Chapter 6 DISCUSSION AND CONCLUSIONS

The results of the analysis are discussed, and answers to the propositions and the research question are presented. The section also contains a reflection on the representativeness of the research and areas for future studies.

APPENDIX CASE STUDY DATA

In the appendix protocols for data collection and results from the interviews are compiled.

INTRODUCTION

1.6.1 Reading instructions

Readers generally interested in the problem definition and its answers may start with reading chapter 1 and 6. For descriptions of the theoretical framework and its evolution, and application of the theories in the analysis, chapter 3 and 5 can be read. Further are the underlying methods for the research in general and for approaching the cases described in chapter 2 and 4, along with a more detailed presentation of the projects in the case study. How the chapters are related is described in Figure 1:1.

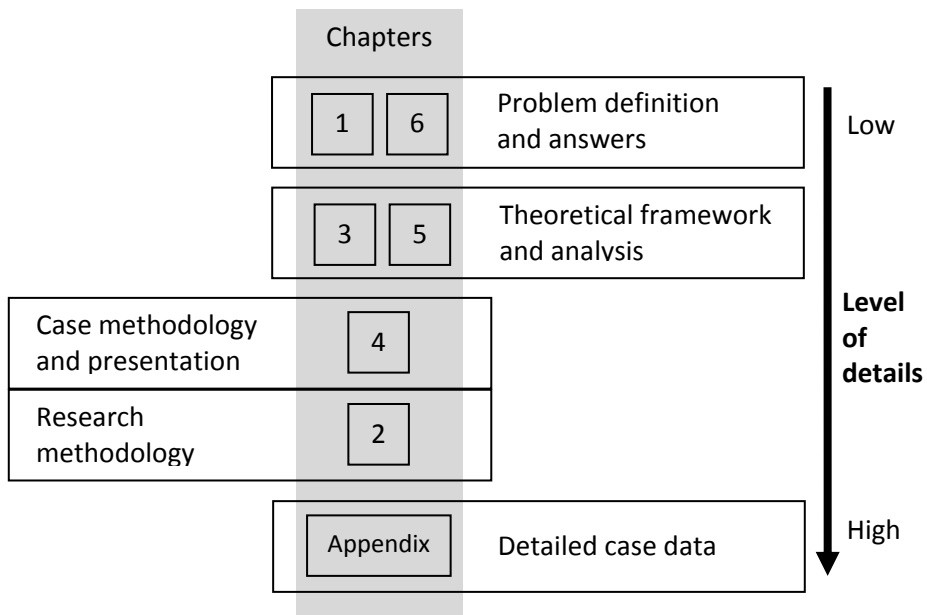


Figure 1:1 Relation of the chapters

2 **METHODOLOGY**

This chapter presents various general methodological approaches to scientific research, and discusses which one is in use in this study. Particular strategies for case-study design and methods for data collection and analysis are also presented. The chapter ends with a description of the practical approach to the study and an analysis of the sources of criticism.

2.1 **RESEARCH PURPOSE**

Different methodologies for performing a study should be used for different types of studies. Four major types of studies are to consider, classified by the purpose of the research (5 p. 29):

- *Descriptive studies* have as main purpose to describe how something works or is performed.
- *Exploratory studies* are in-depth research, aiming to gain understanding of how something works or is performed.
- *Explanatory studies* aim to find cause explanations to how something works or is performed.
- *Problem solving studies* have as purpose to find a solution to an identified problem.

2.1.1 **Research purpose – approach and considerations**

The agile project methodology analyzed in this study – Scrum – is in general well described in literature. The study's purpose is as a first step to identify underlying principles creating customer value when using the agile methodology, meaning that the study is exploratory. As a second step, these findings are used to benchmark the Mobical and other projects in a descriptive way.

METHODOLOGY

2.2 RESEARCH STRATEGY

2.2.1 Research methods

A research strategy should be selected considering the purpose of the study, and used consistent through the whole study. The four most relevant methods when writing a master's thesis are described by Höst, Regnell and Runeson (5 p. 30) and by Yin (6 p. 5):

- *Surveys* are used for description of the current situation for a studied object or phenomenon. Surveys are relevant for answering research questions of form "who", "what", "where", "how many" and "how much".
- *Case studies* are in-depth studies of one or several cases, where the researcher acts as an observer and should affect the studied object as little as possible. Case studies are relevant for answering research questions of form "how" and "why".
- *Experiments* perform a comparative analysis of two or several alternatives, where a few factors are isolated and manipulated one at a time. Experiments are relevant for answering research questions of form "how" and "why".
- *Action researches* are carefully monitored and documented studies of activities, which aim to solve specific problems. The strategy is used for improving an activity at the same time as it is being studied. Action researches are relevant for answering research questions of form "how" and "why".

What strategy to use depends mainly on the type of research purpose and question and the extent of control an investigator has over actual behavioral events (6 p. 5).

2.2.2 Fix and Flexible methodologies

A methodology can be *fix* or *flexible*, depending on the possibilities to adjust the study as it is performed. In a fix methodology, the study is mainly defined prior to the implementation (e.g. surveys and experiments). In a flexible methodology, the study is allowed to be continuously adapted as the conditions change (e.g. case studies and action researches) (5 p. 31).

METHODOLOGY

2.2.3 Quantitative and Qualitative research

Data that is collected can be classified as quantitative or qualitative (7 pp. 190, 191) (5 p. 30), and described as follows:

- *Quantitative research* means to measure how much or how many, with the purpose to explain. Quantitative data can be processed with statistical analysis.
- *Qualitative research* means to get understanding and describe the type and nature of a phenomenon, and qualitative data can be analyzed with sorting and categorizing methods.

For complex problems it is preferable to use combinations of quantitative and qualitative data.

2.2.4 Research strategy – approach and considerations

This research is studying a process through a benchmarking approach, in a context where a few sources of information represents the business segment. The research aims to answer questions of the form “how”; in what way Scrum can be lifted from an internal management tool to get the customers more involved, and how to classify customers.

The research is in general built on a flexible methodology approach, making it possible to adapt the study for the specific conditions at each studied project in the benchmark. This is not possible with a fix methodology. Though, to get comparable measures, the fix methodology approach is applied to high-level comparison of the studied projects.

The purpose of the study is both exploratory and descriptive, and it aims to gain in-depth knowledge about customer relations and processes, related to one specific project. Together with the demand of a mainly flexible approach, the qualitative method is most appropriate in this study.

In light of the discussion above, the case study method is found to be the most appropriate for this research. The survey method, which requires that a sample can be made of a larger population for analysis of quantitative data, is not applicable, and since there are limited possibilities to manipulate the studied object during the research, neither are experiments or action research.

The case-study method deals with the need of in-depth penetration, a flexible approach and qualitative data analysis, and the method supports the purpose of this research well.

METHODOLOGY

2.3 *CASE STUDIES*

Yin (6) underlines the importance of the case study as a research method, and presents a set of systematic procedures for attaining validity and reliability in the research. His approach to case studies is applied to this research by the following procedures.

2.3.1 **Research design**

To be able to carry out the case study, a plan – a *research design* – is needed. Yin describes that “the design is the logical sequence that connects the empirical data to a study’s initial research questions and, ultimately, to its conclusion” (6 p. 20).

The research design consists of five major components:

- *Study questions*, or research questions, clarify the nature of the study, and are most often of the form “how” and “why”.
- *Study propositions* direct attention to something that should be examined within the scope of the study. The propositions describe what you really are going to investigate, and could also be explained as hypotheses. Exploratory studies, though, are not having any hypotheses. Instead the purpose of the study should be stated along with the criteria by which an exploration will be judged successful.
- *Units of analysis* define what the ‘case’ is, e.g. one or several individuals or organizations.
- *Linking data to propositions* describes what methods are used to draw conclusions, and gives an initial approach for the analysis.
- *Criteria for interpreting the findings* give guidance for estimating relevance in the collected data and results.

The research design components lay a solid ground for the topic being studied, and help the researcher to construct a preliminary theory prior to any data collection. They also provide strong guidance in determining what data to collect and strategies for analyzing it.

2.3.2 **Case study designs**

When designing case studies, a distinction can be made between single- and multiple-case designs. As a part of the research design, the researcher has to decide which one to use to address the research question.

METHODOLOGY

The single-case design is studying one unique case, and could be applied when a single case represents a critical case in testing a well-formulated theory, to an extreme or unique case, or to a representative or typical case. Single-case designs involve a risk of misrepresentation, and require careful investigation of the potential case. Properly defining the units of analysis ensures that the case is relevant to the addressed issues.

Applying a multiple-case design, i.e. studying two or more cases, often makes the evidence considered more solid. Though, each case should serve a specific purpose within the overall scope of the research, and be selected either to predict similar or contrasting results to strengthen the evidence for the theory. A multiple-case design could start with the study of a pilot case, from which the researcher can improve his methods for collecting the proper data. The pilot case study should not be considered as a pretest, but the results should be as important as from the following studies in the analysis.

2.3.3 Case studies – approach and considerations

To create sustainable conditions for performing this case study with high measures of validity and reliability, it is chosen to be constructed on Yin's research design. The five components of the research design are described below.

Applying a single-case design studying the Mobical project team may provide evidence for the developed theory. Though, to strengthen the evidence, and being able to benchmark with competitors in the industry, the multiple-case design is applied to this research. The Mobical project team serves as a pilot case, providing the opportunity to refine models and methods for collecting data.

Study question

The study question for this research is earlier described in the introduction of this report: *How and where can Scrum be lifted from an internal management tool to get the customers more involved, and where is customer value created?*

Study propositions

This study's propositions, purpose and the criteria by which it will be judged successful are initially presented above in 1.4 PROBLEM DEFINITION AND PURPOSE, and further described as follows. The first two propositions are

METHODOLOGY

theoretical and describe agile development in general. The following two are specifically related to Mobical and Tactel's business context.

Which drivers of customer value can be identified in agile practices? The study is judged successful if:

- Drivers of customer value are identified
- The study indicates a prioritization of the drivers

How can customers be classified in matter of participation in an agile project? The study is judged successful if:

- Customer types are identified
- Relations between the customer types are clarified

How well are the drivers and customer types considered by companies in Tactel's business context? The study is judged successful if:

- The benchmark indicates how aware the studied projects are of how they relate to different types of customers
- The benchmark indicates how aware the studied cases are of the agile methodology, and how well it is practiced

How can Mobical benefit from the findings? The study is judged successful if:

- Mobical's customer types are identified
- Differences and similarities between the cases are shown
- The study identifies which drivers Mobical should work with to increase customer value and gain business value

Sub-questions for these four areas clarifies what to investigate, and are further analyzed in the subsequent chapters.

Units of analysis

Since the study is of multiple-case design, the studied projects are the units of analysis. These are described in detail in chapter 4.

Linking data to propositions

Literature studies and identification and development of a theoretical framework, give in chapter 3 preliminary answers to the first two propositions.

METHODOLOGY

The data collection, with details described in 4.1 DESIGNING THE CASE STUDY QUESTIONS, is constructed to provide evidence and collect proper information for answering the propositions. General data collection principles and considerations are described below in 2.4 DATA COLLECTION.

Criteria for interpreting the findings

The preferred analytic strategy for this study is to group the empirically collected data and link it with patterns identified in the theories, and in a second step to compare the different cases. This approach both provides evidence for the developed analysis models and answers to the Tactel-related propositions and the main research question.

2.4 DATA COLLECTION

Yin describes three principles to guide the collection of data (6 pp. 97-106), presented below. Following the principles makes the process explicit and ensures quality control.

- *Use multiple sources of evidence.* By using multiple sources of information to address the same facts, findings and conclusions are likely to be more convincing and accurate. Different types of sources are described below.
- *Create a case study database* for organizing and documenting the data. In this way any other person can access the raw material, which increases the reliability of the case study.
- *Maintain a chain of evidence.* To increase the reliability, it should be clear to follow the derivation of any evidence from the report – from the initial research questions via collection of data to the study conclusions.

Especially in multiple-case designs, *case study protocols* could be used when collecting data. The protocol should, according to Yin (6 pp. 67-69), contain four sections:

- An overview of the case study project
- Field procedures
- Case study questions
- A guide for the case study report

METHODOLOGY

When collecting data in a case study, some different techniques and types of information sources are commonly used (5 p. 34); interviews, observations and written documentation. The techniques are described below.

2.4.1 Interviews

Interviews are one of the most important sources of case studies (6 p. 89), and can be classified as (5 p. 34):

- *Structured interviews* correspond more or less to oral surveys, and are based on pre-defined lists of questions that are to be followed to the letter.
- *Semi-structured interviews* are constructed upon a set of questions, used as support for the interview. This kind of interview brings structure to the collected data, but enables an open discussion.
- *Unstructured interviews* let the interviewee lead the conversation and speak freely. The interviewer's role is to make sure that the interview sticks to its topic.

Interviews can provide important insights into a situation, but are subject to common problems of bias and poor recall.

2.4.2 Observations

Observations mean to study events and behaviors, either as a participating observer or as a direct, passive, observer. The observations can be made formally and well documented in protocols etc, or less formally, for example at a field visit. Observations are often useful to provide additional information to a studied topic.

2.4.3 Written documentation

Going through documentation, written for another purpose than the study being performed, is a common and relevant way to provide information, and documents play an explicit role in any data collection.

2.4.4 Data collection – approach and considerations

In this research data are collected from multiple cases, i.e. multiple projects in the industry. Within each case are different sources of information considered; primarily interviews with product owners and project managers, but also written documentation such as company web sites. Since agile teams have flat,

METHODOLOGY

non-hierarchical organizational structures, it is not meaningful to collect data from sources at different levels to vertically penetrate the studied cases. Instead the cases are studied in a horizontal approach, by at least two different sources of information within the same case. In all cases, though, data is collected from project members with different responsibilities, where some are expected to have deeper knowledge about customer relations, and others about internal and technical processes.

The collected data is organized in a database, presented in APPENDIX CASE STUDY DATA. This report is also, as described in the subsequent chapters, linking the research questions to the theory, the theory to the collection of data, and the data to the analysis and conclusions. All this is done to keep up a high level of quality of the evidence in matters of validity and reliability.

This report serves as a case study protocol itself, in which an overview of the case study and a guide for the report are described in chapter 1, field procedures are described in chapter 4, and case study questions in APPENDIX CASE STUDY DATA.

To get information and background material for the theoretical framework, written documentation such as books and journal articles are generally considered. As described above, interviews and written documentation are used for collecting data for the cases. The need for comparable and qualitative data makes the semi-structured interview an appropriate approach; it supports a set of questions identical for all cases at the same time as the interviewee can speak rather freely.

In some cases, especially the Mobical case, observations provide additional information. The observations are made in less formal ways mainly by direct observations.

METHODOLOGY

2.5 PRACTICAL APPROACH

This sub-chapter presents in a summarized way how this master's thesis was performed in a practical manner, and it describes the following phases:

1. Startup and planning
2. Problem definition
3. Literature study
4. Research design
5. Data collection and Analysis
6. Discussion and conclusion

The phases, their mutual relationship and how they are related to the disposition of this report are illustrated in Figure 2:1. In the figure, the main flow is shown with thicker arrows.

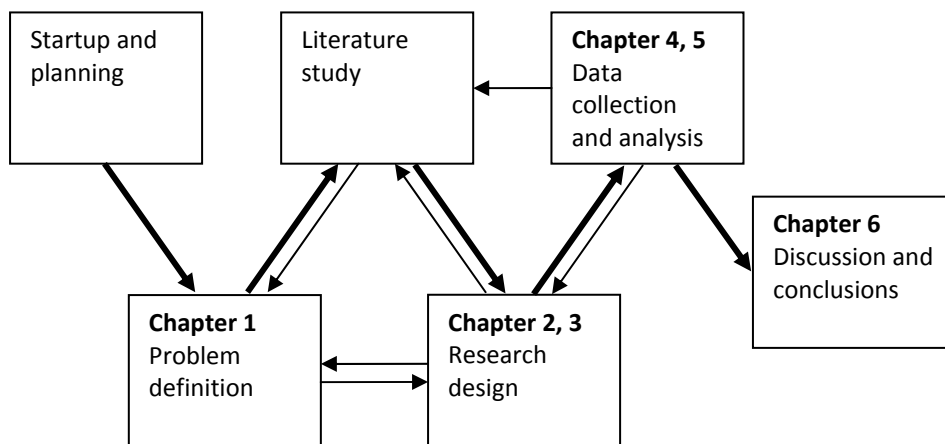


Figure 2:1 Relation between the research development phases

2.5.1 Startup and planning and Problem definition

During these phases the goals and problem definition for the research was defined, and the project was planned. The subject for research was approved by both the Department of Industrial Management and Logistics and by Tactel AB, and a project specification was developed.

METHODOLOGY

2.5.2 Literature study

After defining the area of research, relevant literature was searched. Books, journal articles, conference proceedings and web forums within the areas of project management and software methodology were consulted. This initial general orientation of the topic led to a concrete literature search.

The following search engines, databases and web portals were mainly used in the search:

- ELIN – university data base
- <http://www.scirus.com>
- <http://apm.org.uk>
- <http://www.pmi.org>

In the search the following keywords were used:

- agile
- agile + customer
- agile + customer + value
- customer + value

The search resulted in a list of articles, conference proceedings and books, which were sorted by relevance for this research. Great literature input was also given from the supervisor. By consulting the references in the collected literature, the list grew even longer.

All literary sources were reviewed with respect to reliability and validity for this research, and the most relevant have made the foundation for the theory. The reference list at the end of this report contains the complete list of literature used in this research.

2.5.3 Research design

Based on studies of methodological literature the design of the research was created, and the development of the theoretical framework was part of this phase of the research. The research design is described above in 2.3.3 Case studies – approach and considerations.

METHODOLOGY

2.5.4 **Data collection and Analysis**

As a step in developing the research design, the units of analysis were identified. A number of projects were found to meet the criteria, described in 4.2 PERFORMING THE STUDIES, and approached for further studies and interviews. The collected data was organized in the case study database, presented in APPENDIX CASE STUDY DATA, and analyzed in light of the theories, as visualized in Figure 4:1 and Figure 4:2 on page 48.

2.5.5 **Discussion and conclusions**

Conclusions were drawn from the analysis, and the results were put together in this report. They are also to be presented both at a seminar at Tactel and at a public seminar at Lund University.

2.6 *SOURCES OF CRITICISM*

2.6.1 **Research quality**

Research quality can be judged from a number of aspects, which the researcher has to consider throughout the whole research process. (5) (7)

- *Relevance* describes if the achieved results are relevant in the given context and if the data collected is usable.
- *Reliability* means how trustworthy a source, method or analyze is.
- *Validity* describes the relevance of the collected data and the sources of information corresponding to the formulation of problems and goals.
- *Objectivity* stands for an open-minded approach without bias from the researcher.
- *Creditability* means if the generated knowledge is tenable.
- *Representative* results mean that the generated knowledge is representative for the area of inquiry.
- *Transferability* describes whether the results also can be generalized for other contexts than of the specific study.

The quality aspects can be increased by the use of triangulation, which means to use several methods, data sources or theories for examining the same object (6 pp. 97-99).

METHODOLOGY

For establishing quality in case studies, Yin presents four tests to be considered in different phases of the study (6 p. 34):

- *Construct validity*: establishing correct operational measures for the concepts being studied. Occurs in the data collection phase.
- *Internal validity*: establishing a causal relationship, whereby certain conditions are shown to lead to other conditions, as distinguished from spurious relationships. (Not for descriptive or explanatory studies.) Occurs in the data analysis phase.
- *External validity*: establishing the domain to which a study's findings can be generalized. Occurs in the research design phase.
- *Reliability*: demonstrating that the operations of a study – such as the data collection procedures – can be repeated, with the same results. Occurs in the data collection phase.

Based on several sources, Höst and Runeson (8) have derived checklists supporting researchers in conducting software engineering case studies. The resulting checklists, paying much attention to structure and validity, focus on five areas within the case study:

- Case Study Design
- Preparation for Data Collection
- Collecting Evidence
- Analysis of Collected Data
- Reporting

2.6.2 Quality – approach and considerations

In this study, the quality aspects are considered in all stages; from the initial research questions, via literature studies and data collection, to the analysis and conclusions. Using Yin's research design model, the quality issues are well considered throughout the research. By using the model throughout the research process, factors that may have a negative impact on the quality are identified. How they are dealt with is discussed below.

Young science needs careful review of sources

Agile software development methodologies are very young, and yet not much research has been made within the area. Höst and Dybå (1 p. 851) conclude that "the strength of evidence in the current review regarding the benefits and limitations of agile methods, and for decisions related to their adoption, is very low." All previous research considered in this study is carefully reviewed, to

METHODOLOGY

avoid lack of reliability. The most recent evidence for agile method benefits are often published on web forums, but seldom reviewed as rigorous as articles published in journals. Despite the fact that the journal articles not express the very most recent observations, they are considered having higher reliability and hence are preferred as input for this research.

Case study criticism

Case studies as research strategies are sometimes being criticized for their lack of being valid and rigorous, compared to other strategies. This may be because investigators not always have followed systematic procedures and may have allowed biased views in their case studies, which is less likely to occur using other strategies like experiments and surveys (6 p. 10).

By following the systematic procedures, maintaining a chain of evidence and considering Yin's four tests and Höst and Runeson's checklists, as described above, the lack of reliability and validity is minimized in this research. Though, using a quantitative survey method for approaching this research could have gained higher levels of quality in matter of objectivity and reliability, and a wider selection of cases could have made the study further representative.

Quality in data collection and analysis

Quality aspects regarding the data collection are further described in chapter 4. The main considerations concern the use of well defined criteria for selecting the cases, the use of triangulation (both by the multiple-case design and by different sources of information within each case), and making the raw material accessible for others via the case study database. The structured data is also an important tool in this study for linking the data to the analysis and conclusions with retained quality.

Limitations of the research

In some areas the quality aspects lead to limitations of this research, mostly affecting its transferability. Where limitations are identified they are described in the corresponding section of the report, and finally summarized in chapter 6.3. The main limitations concern narrow selection of cases to study and presumed bias in the data collection.

3 THEORY

This chapter presents theories and models in the area of the current research. The theories are a foundation for the analysis of the collected data, and link the research questions to the analysis.

Initially are agile methods and Scrum presented, followed by theories about customer relations and a classification of customers in Scrum projects. Then the concept of customer value is explained and drivers of customer value in agile projects are identified.

The chapter repeats to some part what is outlined about agile methods in chapter 1, with the purpose to make the more extensive descriptions below easier to follow.

3.1 AGILE METHODOLOGY

Agile methodologies provide techniques for delivering customer value by producing higher quality software in a shorter period of time. Erickson, Lyytinen & Siau (9 p. 89) define the meaning of agility as “to strip away as much of the heaviness, commonly associated with traditional software-development methodologies, as possible to promote quick response to changing environments, changes in user requirements, accelerated project deadlines, and the like.”

As presented in chapter 1, various agile methods exist. The principles common for all of them are stated in the Manifesto for Agile Software Development (2), hereafter called the agile manifesto. The manifesto outlines four core values:

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

That is, while there is value in the items on the right, the items on the left are valued more. This means for example that processes and tools are important, but individuals and interactions are more important in the software development process.

THEORY

3.1.1 Basics of agile methodology

The different agile development methodologies have some common basic characteristics, described by Augustine (3 p. 21) as follows.

- *Small releases.* Work is divided into small packages to manage complexity and to get early feedback. Releases are usually delivered in one to three months.
- *Iterative and incremental development.* Plans, requirements, design, code and tests are evolved incrementally through multiple iterations, rather than through a single “waterfall” pass.
- *Co-location.* All team members are collocated to facilitate face-to-face-communication.
- *Release plan/feature backlog.* Desired features are defined at a high level and prioritized by customers in a release plan or feature backlog. The prioritization is done collaboratively with the developers, which also provide effort estimations.
- *Iteration plan/task backlog.* Lower-level features are defined and prioritized at the start of each iteration. Developers provide effort estimations and customers decide business priority.
- *Self-organizing teams.* The team is collectively responsible for allocating and completing the backlog tasks, without top-down management control.
- *Tracking.* The progress of the features and tasks being developed are tracked within an iteration, showing the relation between finished and remaining work. The tracking is done by the team members and increases visibility of the project status for both the team and stakeholders.
- *Simple, lean and adaptable.* All aspects of work, including processes, are kept simple, lean (low on wastes), and adaptable to maximize customer value and to accommodate change.

The various agile methodologies may differ in practices for implementing the characteristics. Since the focus of this study is on Scrum, as described in chapter 1.5, the other methodologies are not further explained in this report.

3.1.2 Benefits and limitations

Agile methods are considered to be superior to traditional development processes when a project can take advantage and gain value from their benefits. Dybå and Dingsøy (1) have, in their systematic review of empirical studies of agile software development, investigated what is currently known

THEORY

about the benefits and limitations of agile methods. Their findings are explained and expanded as follows according to Schwaber (10), (11) and Korkala, Abrahamsson & Kyllönen (12).

Benefits with agile methods are found in the following areas:

- *Improved customer collaboration.* Customers are satisfied with the opportunities for feedback and responding to changes.
- *Increased visibility.* The progress is easy to follow for stakeholders.
- *Reduced time to market* by small and high frequent releases.
- *Increased value to market.* Change requests from customers are always accepted, and most valuable features are developed first.
- *Improved work process for handling defects,* including faster solving bugs and decreased defect rates.
- *Improved estimation* of time and cost.
- *Easy adoption.* Agile development practices are easy to adopt.
- *Improved learning* by exercises as pair programming.

Limitations and criticism are identified within the following areas:

- *Design and architecture.* The lack of attention to design and architectural issues may lead to suboptimal design-decisions.
- *Skilled teams.* The importance of staffing agile teams with people that have faith in their own abilities combined with good interpersonal skills and trust.
- *Large teams.* Agile development methods are more suitable for small teams, since it may be difficult to introduce agile methods into large and complex projects.
- *Scientific support.* There is little scientific support for many of the claims made by the agile community.
- *The practices in XP* are not often applicable, and are rarely applied by the book.
- *Agile development is nothing new,* and such practices have been in place in software development since the 1960s.

3.2 SCRUM

The word *scrum* is originally a rugby term for a close formation in which the team is collaboratively working to win the ball. The software methodologists Ken Schwaber and Jeff Sutherland picked up this term, and described in the

THEORY

mid 1990s Scrum as a process for software development. Since then, Scrum has become one of the most practiced agile methods. This sub-chapter presents Scrum as it is described by Schwaber (11 pp. 101-112) (10).

3.2.1 Scrum overview

The base for Scrum is the *sprint* – focused work against agreed goals in an iteration of two to four weeks. The output of each sprint is a deliverable increment of product. During the sprint a daily inspection – *Daily Scrum* – is held, where the team members meet to inspect each other's activities and synchronize their work. A list of requirements – the *Product Backlog* – states requirements for the entire project, and for each sprint the highest prioritized requirements in the backlog are transferred to the *Sprint Backlog*. Figure 3:1 gives an overview of the Scrum process.

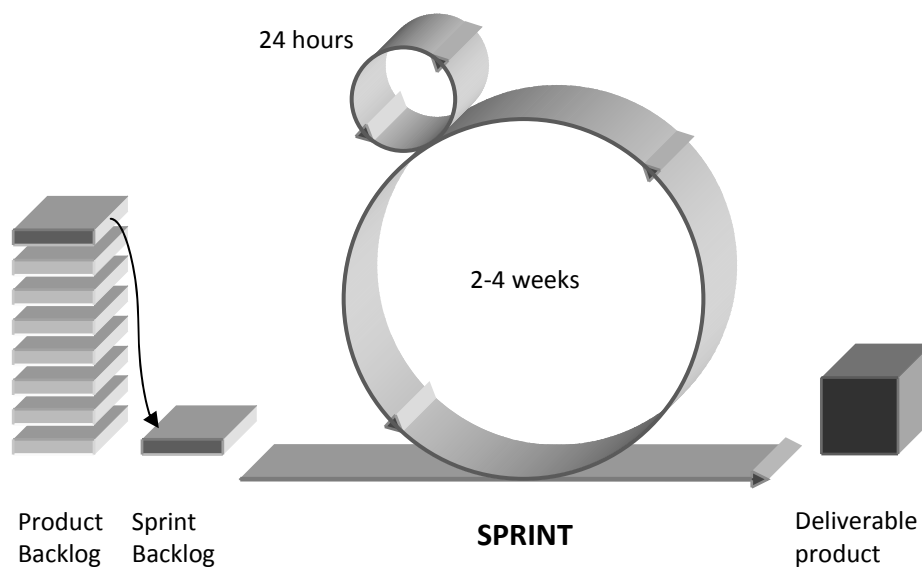


Figure 3:1 Scrum – an overview

At the start of each sprint, the team selects which requirements it believes it can turn into an increment of deliverable functionality by the end of the sprint. Within the sprint, the team organizes itself with regard to skills and capabilities, collectively plans how to fulfill the requirements, and modifies the approach daily. At the end of the sprint, the team demonstrates the functionality to project stakeholders.

THEORY

3.2.2 Three Roles

There are only three roles in the Scrum process between which all management responsibilities are divided:

- *The Product Owner* represents everyone with a stake in the project, achieves funding and administrates the Product Backlog. He is responsible for frequently prioritizing the backlog, so the most valuable functions are produced first. The role of the Product Owner could be held by a person in the customer or in the vendor organization.
- *The Scrum Master's* role is to coach the team, and remove any barriers between development teams, Product Owner and customers. He is responsible for the Scrum process, for teaching it to everyone involved in the project, for implementing it, and ensuring that everyone follows its rules and practices.
- *The team* normally consists of 5-9 people, and is developing the functionality by being self managing, self organizing and cross functional. The team members are collectively responsible for the success of each iteration and the project.

3.2.3 Flow

A Scrum project starts with a vision of the product being developed, which is translated to high level requirements and a baseline plan of cost and time frames. The Product Owner creates the Product Backlog – a list of requirements put together taking into account the vision and the demands on return on investment. The items most likely to generate value are given top priority in the Product Backlog, which is reviewed frequently during the project to meet changes in market demands and customer prioritizations.

Each sprint starts with a Sprint Planning meeting, where the Product Owner and the team decide what will be done for the next sprint. The team plans out the sprint and defines the Sprint Backlog by breaking down the high-level requirements from the Product Backlog. The responsibilities within the scrum flow are illustrated in Figure 3:2.

Every day the team gets together for the Daily Scrum meeting, with purpose to synchronize the work of all team members. Each member reports what is done since last Daily Scrum, what he plans to do until next Daily Scrum, and if any obstacles are in the way for his work.

THEORY

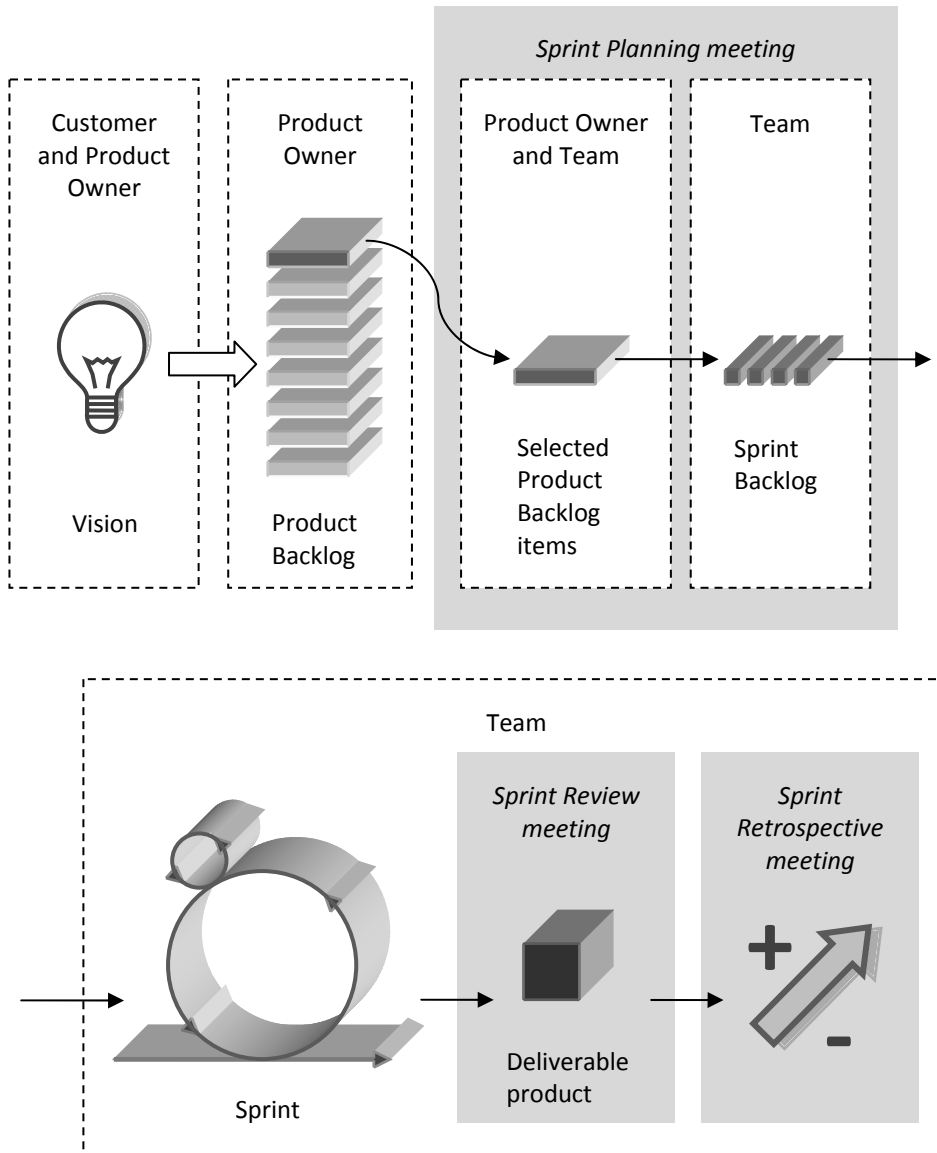


Figure 3:2 Responsibilities in the Scrum process

At the end of each sprint a Sprint Review meeting is held, where the team demonstrates what has been developed for the Product Owner and stakeholders. Customers, marketing department, senior management and others could be counted as stakeholders. Also a Sprint Retrospective meeting is held where the team revises its development process.

THEORY

The cycle of sprints repeats until the project is no longer funded. The longer a project is funded, the more items from the Product Backlog are realized, but since a deliverable increment is produced in each sprint, there will always be a functioning product no matter when the funding ceases.

3.2.4 Scaling

The optimal size for a Scrum team is seven people, give or take two. On larger projects Scrum is scaled to comprise multiple Scrum teams, and hundreds of people could work according to Scrum in a single project. The usual mechanism that coordinates the multiple teams is called a *Scrum of Scrums* (10 p. 44), which is a Daily Scrum meeting consisting of one member from each team. For this multiple-team technique to be effective there should be only minor dependencies and couplings between the teams that require resolution. The multiple teams work with the same Product Backlog, but with different Sprint Backlogs, as shown in Figure 3:3.

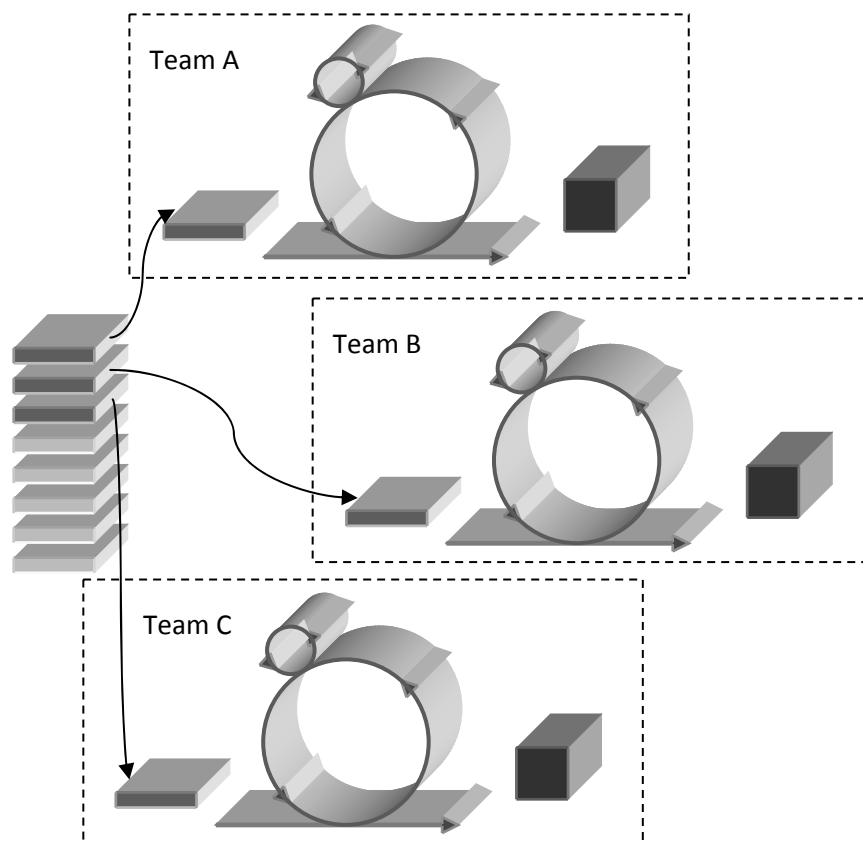


Figure 3:3 Multiple Scrum teams

THEORY

A scaling infrastructure must be put in place prior to scaling any project (10 pp. 122-123), addressing mechanisms for synchronizing the teams, detailed architecture and project environment. The development of the infrastructure is done in sprints, and only one team can sprint until it is in place. When in place, the coordinating of multiple teams can start. It is important to optimize the capabilities of the initial team, and then staff the additional teams with at least one member of the initial one. The initial team members will become the new teams' experts on infrastructure and architecture.

3.3 THE CUSTOMER AND SCRUM

Based on how Schwaber (10) describes the customer role, two ways for the customers to be part of the Scrum process are identified; direct or indirect. Their degree of involvement can also be classified as high or low, as shown in what in this research is referred to as the *Scrum Customer-classification System* (SCS) in Figure 3:4.

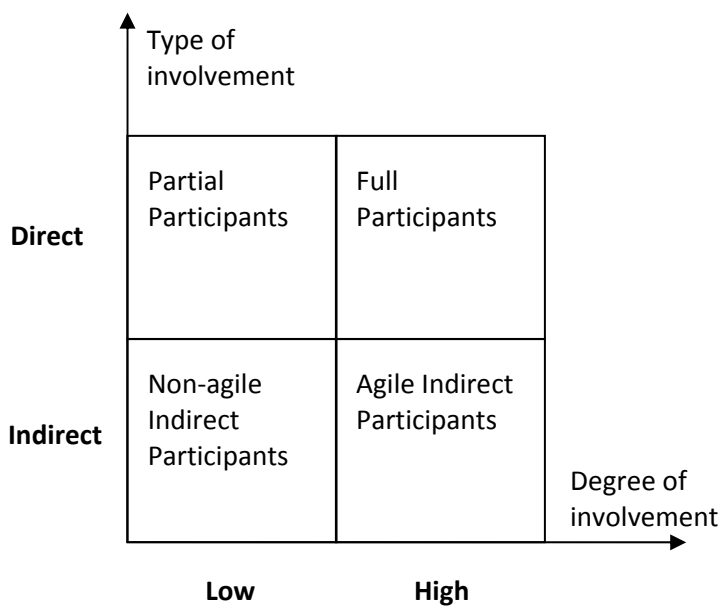


Figure 3:4 Scrum Customer-classification System (SCS)

In SCS customers are allocated to any of the four quadrants to visualize their involvement in the Scrum process. Actual customers will not necessarily be perfectly aligned with any of the four groups, and a customer's degree of involvement can for example be moderate instead of high or low. Though, the

THEORY

visualization of a customer's position can be used when considering strategies for evolving business relationships. For example, if a customer is identified to be a Partial Participant, business relations can be strengthened by transferring the customer against the Full Participant quadrant.

The hierarchy of the levels in SCS and the normal way to progress through the matrix is shown in Figure 3:5.

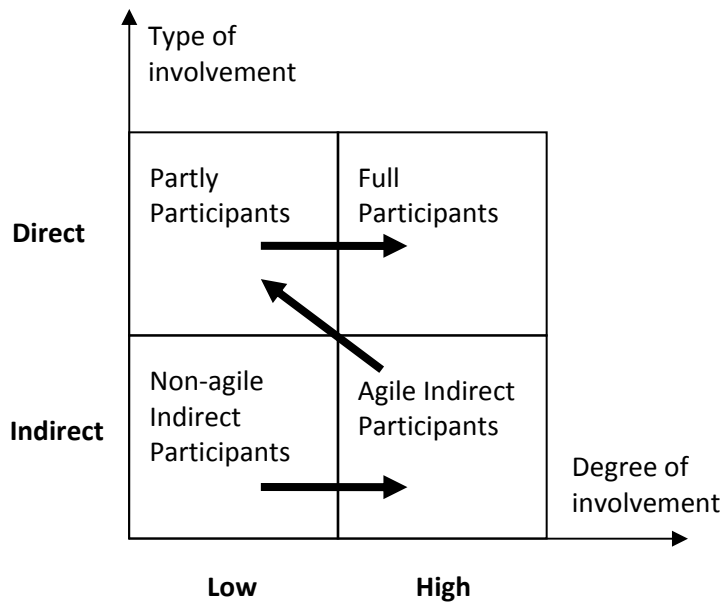


Figure 3:5 SCS hierarchy

When deciding what actions to take, the customer's position must first be known, and for this the SCS is to be used. The four types of customer involvement are presented in detail below.

3.3.1 Direct participant customers

Direct participant customers are holding the role of the Product Owner in projects, and drive the development by owning and prioritizing the Product Backlog. They are allowed to attend any meeting within the Scrum process and take part of the sprint review. This approach is applicable when a project is driven by a single customer, or by a single major customer for which all other, minor ones and their requirements are well known. The result is that the single or major customer is solely involved in the development project.

THEORY

Full Participants

Full Participants are customers entirely acting as project members. They are collocated with the teams, and the transparency of the vendor organization is very high for the customer. The customers are required to have refined Scrum experience and a good knowledge of the product to develop to cope with this role.

Partial Participants

Partial Participants do perhaps not possess a complete understanding of Scrum, and the vendor organization may not be totally transparent. The customers act as product owners – prioritizing the Product Backlog – but an assistant product owner or customer project manager at the vendor company is responsible for the communication with the team and internal stakeholders.

3.3.2 Indirect participant customers

Indirect participant customers do not possess any of the three Scrum roles. Instead the Product Owner often is an employee at the vendor company representing one or multiple customers, and prioritizing all their requirements. In those projects the customers can be more or less involved as described below. The transparency is lower for the indirect participant customers, making it possible for the vendor company to prioritize different customers in non-equal ways, with purpose to gain business advantages.

When working with indirect participant customers the agile methodology mostly is used as an internal tool. Projects with combinations of both agile and non-agile indirect participants may exist.

When multiple customers exist for a project, one Product Backlog can be maintained for each customer. These multiple backlogs are then merged to a central one, where the Product Owner prioritizes between the different customers' requirements (10 pp. 126-129), as shown in Figure 3:6.

Agile Indirect Participants

Agile Indirect Participants are aware that their projects are developed with agile methodologies, how long the sprints are, and how the frequent requirement prioritizations and deliveries work. The customers are involved in regular meetings to re-prioritize the Product Backlog, may sometimes attend the demos, and in some cases even the Scrum meetings. In multiple-customer

THEORY

projects, the customers often are aware of their role as one of several customers.

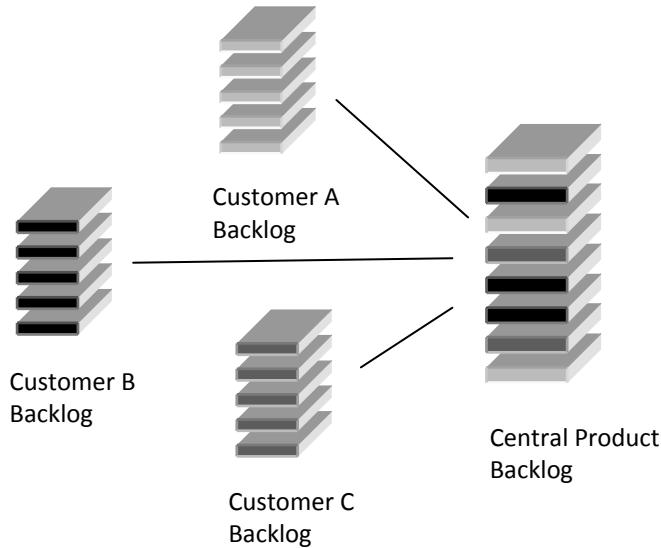


Figure 3:6 Backlog and multiple customers

Non-agile Indirect Participants

Non-agile Indirect Participants do not need to be aware that their product is developed with an agile methodology. They are either buying an existing product or are specifying the requirements at start of the project in a traditional way. The customers may be aware that functionality is developed in increments, but do not participate in regular reviews of the Product Backlog.

Existing products may also be customized to suit each customer's specific needs. In those cases the Product Owner at the vendor company is responsible both for providing value in the base-line product and for prioritizing the different customer's adaptations.

The approach with Non-agile Indirect Participants may work well for smaller projects and in projects with low customer-specific complexity.

3.4 *CUSTOMER VALUE*

Augustine (3 p. 20) refers to the classic definition of customer value from Lean Thinking when he states that agile methodologies, compared to traditional ones, offer a viable alternative to increase customer value; the right product for the right price at the right time. The right product is the product with exactly the features that the customer wants, the right price is what is believed to be a fair deal, and the right time is when the customer wants it.

Khalifa (13) reviews and synthesizes literature on the subject of customer value, and his research is referred to below to give a broader explanation of different interpretations of the concept of customer value. Initially, he finds a generally valid agreement that *customer value is determined by customers' perception and not by suppliers' assumptions or intentions*. This means that it is the customer that really determines to what extent value is perceived – not the vendor.

He further sorts his findings of models describing customer value in three categories, where each category emphasizes different dimensions of customer value:

- Value components models
- Benefits/cost ratio models
- Means-ends models

The views of customer value represented by each category are described in the following sub-sections. Customer value for a specific case can be described from either of the category perspectives, and from combinations of them.

3.4.1 **Value components models**

Khalifa describes the characteristics of Value components models by Kano's model of customer perception, where value is represented as the performance or physical characteristics of a product. It includes three components which focus on customers' benefits:

- *Dissatisfiers* – characteristics that are taken for granted and must be present, and bring customer satisfaction to “normal”. Their absence annoys the customers.
- *Satisfiers* – features that are expected and explicitly requested by the customers. Their absence disappoints the customers.
- *Delighters* – characteristics that the customers do not expect, and which solve latent needs of the customers. They have no negative effect if absent, but only a positive when present.

THEORY

The value components models are useful when developing new products or services and considering new features, but pay modest attention to relations between customer and vendor. They also focus on customer's benefits without considering the sacrifices.

3.4.2 **Benefits/cost ratio models**

Customer value is the difference between *total benefits* and *total sacrifices*; what is received in form of quality, profit and worth versus what is given up. The sacrifices could be monetary, time, search and learning costs, as well as cognitive and physical effort coupled with financial, social and psychological risks. These utilitarian models are more complete and broader than the value components models, and consider customer value in a longer time horizon perspective.

3.4.3 **Means-ends models**

Value is the accomplishment of favorable *ends* through the use of a product as the *means*. Value is created when a product lives up to consumers' personal values, mental images or cognitive representations underlying their needs and goals. These models explain why customers attach different weights to various benefits in evaluating alternative products and services, and take into account the negative consequences of certain product or service attributes.

3.5 *AGILE DRIVERS OF CUSTOMER VALUE*

To identify areas where agile methods specifically have impact on the customer value, theories and findings from various literary sources are presented below. Initially Chow and Cao's (14) study of critical success factors of agile software development projects is described, which then is reinforced and expanded with findings of other sources. At the end of this sub-chapter, specific *drivers* of customer value in agile projects are identified, based on Chow and Cao's expanded theory. The drivers can be used when analyzing the agile process in a project and how it creates customer value. Finally, the drivers are mapped to the four customer types in SCS, and this mapping can be used in the analysis to point out areas to consider increasing value for different types of customers.

THEORY

3.5.1 Critical Success Factors

In their extensive survey, Chow and Cao (14) identify and provide insight into *Critical Success Factors* (CSFs) that help software development projects using agile methods to succeed. They define CSFs as factors that must be present for an agile project to be successful.

In their study, success is measured by four dimensions:

- *Quality* – delivering good products or project outcome
- *Scope* – meeting all requirements and objectives
- *Time* – delivering on time
- *Cost* – delivering within estimated cost and effort

These dimensions correspond to the definition of customer value from Lean Thinking presented above in chapter 3.4 – right product (quality and scope) for the right price (cost) at the right time. This means that the CSFs studied by Chow and Cao relates to success in matter of customer value.

By the four dimensions they identify six CSFs, of which three are classified as significant CSFs and three as auxiliary CSFs. They are described by their attributes and presented as follows in decreasing order, starting with the most significant.

1. Delivery Strategy
 - a. Regular delivery of software
 - b. Delivering most important features first
2. Agile Software Engineering Techniques
 - a. Well-defined coding standards up front
 - b. Pursuing simple design
 - c. Rigorous refactoring activities
 - d. Right amount of documentation
 - e. Correct integration testing
3. Team Capability
 - a. Team members with high competence and expertise
 - b. Team members with great motivation
 - c. Managers knowledgeable in agile
 - d. Managers having adaptive management style
 - e. Appropriate technical training to team.

THEORY

4. Project management process
 - a. Following the agile-oriented requirement management process
 - b. Following the agile-oriented project management process
 - c. Following the agile-oriented configuration management process
 - d. Good progress tracking mechanism
 - e. Strong communication focus with daily face-to-face meetings
 - f. Honoring regular working schedule
5. Team environment
 - a. Co-location of the whole team
 - b. Coherent, self-organizing teamwork
 - c. Projects with small teams
 - d. Projects with no multiple independent teams
6. Customer involvement
 - a. Good customer relationship
 - b. Strong customer commitment and presence
 - c. Customer having full authority.

Chow and Cao mean that as long as an agile project implements the three significant CSFs, the project could be likely to be successful; i.e. could be likely to generate customer value. Implementing the auxiliary three may bring further value to the project.

They also identify some common assumptions about agile success factors as non-factors, meaning not critical for a project to succeed:

- *Strong executive support and sponsor commitment*
- *Agile-style work facility* – pair-programming stations, communal areas, wall spaces etc.
- *Organizational Environment* – agile-friendly cooperate culture, oral culture, universal acceptance of agile, appropriate reward system etc.
- *Agile-appropriate project types* – differences between developing software for Internet or intranet usage and traditional computing platforms.

3.5.2 CSFs – Reflection

Success for a project in matter of customer value is determined by the implementation of the CSFs, which are defined by their attributes – underlying practices and methods. This means that the attributes could be categorized as *drivers* of customer value. A driver in this case is a measurable method or practice that when implemented gains customer value, either direct or indirect.

THEORY

Since the CSFs describe what have to be considered for a project to be successful, neglecting one or several CSFs may result in less success and less delivered customer value than expected. The attributes of the CSFs could therefore be categorized as *drivers of dissatisfiers and satisfiers* according to Khalifa's Value components models described in chapter 3.4. Other factors might not be as critical for success, but may bring additional customer value by serving as *drivers of delighters*.

In the following sections, the CSFs and drivers are explained, reinforced and expanded with findings of other sources. The expansion is done to include areas not covered by Chow and Cao, and to make the measurement of the CSFs by its drivers less ambiguous. A compilation of the complete set of drivers and their measures is to be found in chapter 3.5.3 The Drivers. Indexed drivers in the following sub-chapters, e.g. driver 1.A, 2.F etc., refer to the presentation of the drivers in Table 3:2 on page 41.

Delivery strategy

Berteig (15) describes that it is important to consider *frequency of delivery* and *keeping consistent time box* to increase stakeholder engagement and trust in an agile project. This means that short iterations of the same length make it possible to see and compare progress in a project. These findings reinforce and expand driver 1.A Regular delivery of software; *short iterations of equal length*.

Agile Software Engineering Techniques

The importance of following coding standards, simple design, refactoring activities and integration testing is supported by Schwaber (10). He refers to this as *keeping the code clean*. These concepts can be explained as follows:

- *Coding standards* are rules that developers follow to make their code easy to read and maintain.
- *Simple design* means to choose the simplest way to write code for a specific feature.
- *Refactoring* is about removing any duplicate or ill-structured code to make the code simpler.
- *Integration testing* may include test activities in several steps; unit tests, harness tests, and functionality tests.

THEORY

To keep the code clean and for the daily inspection mechanisms to work, Schwaber stresses that all code should be checked in, built and tested every day. He also underlines the importance of the demonstrations at the end of each iteration, and that all demonstrated code should be *complete*. By this he means that it should be written according to the standards, easy to read, refactored, and completely tested. Otherwise the corresponding items in the backlog cannot be considered to be completed.

Taking into account Schwaber's demands on the engineering techniques, two additional drivers are identified: *2.F Daily builds* and *2.G Code completed at demonstration*.

Team Capability

The importance of having team members with high competence is supported by Dybå & Dingsøy (1) and by Elssamadisy (16). They describe that it is important to staff agile teams with people that have faith in their own abilities combined with good technical and managerial skills.

Elssamadisy also states that *sustainable self-discipline* is important for agile teams; to always follow the development process and practices, e.g. following coding standards and build and run automated unit tests. This finding expands driver *3.B Team members with great motivation and self-discipline*.

To measure the managers' roles in teams, the drivers *Managers knowledgeable in agile* and *Managers having adaptive management style* are condensed to one driver: *3.C Managers knowledgeable in agile and having adaptive management style*.

That appropriate technical training to the team is important is supported by Livermore (17). He also highlights the importance of *methodology training* – to provide training of the implemented methodology to the development team. This finding expands the set of drivers with driver *3.E Appropriate methodology training to team*.

THEORY

Project management process

The concepts in some of the drivers in the project management process may need explanation, and are described as follows:

- *Requirement management process* is the way to organize requirements in a Product Backlog, prioritizing them, and selecting the ones bringing most value to be developed in the next iteration.
- *Project management process* describes the breakdown of responsibilities between team and product owner etc.
- *Configuration management process* – the tracking and controlling of changes in the software, tracking of defects, and managing hardware and software that host the system.

The concept of configuration management is ambiguous and could be hard to measure separately. At the same time, what is considered in the configuration management process is also considered as a part of the overall project management process. Therefore it is more justifying to measure the configuration management process as a part of the overall project management process, eliminating the driver *Following the agile-oriented configuration management process*.

The importance of strong communication focus is a common understanding in agile literature, and Korkala, Abrahamsson & Kyllönen (12) underline the importance of face-to-face communication. They even mean that this form of communication should be the default communication method in agile development.

Team environment

With co-location of the whole team means primarily the development team (or teams), but other roles as the Product Owner are also to be considered. The benefits and importance of small teams are supported by Dybå & Dingsøy (1), Livermore (17), and Elssamadis (16). They mean that it is difficult to introduce agile methods to large and complex projects, and that larger organizations have more internal inertia and find it more difficult to implement change than small organizations.

Customer involvement

Good customer relationship and strong customer commitment and presence are supported to be important by Elssamadis (16) and Korkala, Abrahamsson & Kyllönen (12). They underline that an on-site customer expert

THEORY

representative, available full-time or part-time to the development team, enables efficient communication.

Even if efficient communication is enabled, Livermore (17) does not find any significant correlation between successfully implementing an agile methodology and collocating the development team with business customers. Also Martin, Biddle & Noble (18) come in their study of XP projects to conclusion that on-site customers are under stress and commit working long hours, and cannot be sustained for a long period.

Due to the differing findings, the importance of an on-site end customer is unclear and not identified to be a generally valid single driver. Though, the availability of the customer or a customer representative is agreed important, and has to be considered as part of driver *6.B Strong customer commitment and presence*.

Korkala, Abrahamsson & Kyllönen (12) and Mann & Maurer (19) highlight that customers should be trained in the agile process, to better understand expectations from developers and stakeholders. If the customers are not fully aware of the nature of agile development, the importance of communication and feedback can go unnoticed. To measure this finding, the set of drivers is expanded with driver *6.D Customers trained in the agile process*.

Eckfeldt, Madden & Horowitz (20) and Schwaber (10) discuss the limitations of traditional fixed-price and fixed-date contracts in agile projects. They mean that the vendor takes all of the scope risk, and clients commit themselves to a scope too early in the project.

By instead using what Eckfeldt, Madden and Horowitz refer to as *target-cost contracts*, a project's risks and opportunities are shared between the vendor and the customers. A target-cost contract states the initial target scope, estimated time frame and estimated cost for a project, in contrast to traditional contracts where scope, time frame and cost are totally agreed prior to project start.

When using target-cost contracts, the risk that development takes longer than estimated – and the resulting cost – is shared between vendor and customer. This risk creates incentives for the customer to think of ways to reduce cost and complexity.

The target-cost contract also gives the customer the option of canceling the work prior to the contract end date, for example when enough business value

THEORY

has been derived. There would be a penalty for the customer, but it would be less than having the unnecessary requirements developed and implemented.

Trust is the foundation for success in target-cost contracts, and they can only be successful if the customer is open to discussing the target-cost approach. When customers are committing to share the risks and opportunities provided by target-cost contracts, they get more involved in the projects. Therefore is the use of target-cost contracts identified as a driver; *6.E Use of target-cost contracts to share risk.*

3.5.3 The Drivers

The drivers of customer value form a model, which highlights areas to consider improving agile practices and gaining customer value. The identified drivers are not being measured by any fix scale, since such a scale cannot be identified, but are measured relatively to each other. This means that the model cannot be used for a definite comparison between two different projects teams or companies, but for indicating differences and similarities in the projects' agile implementations. Also the underlying practices and processes gaining value for a driver can be used when comparing organizations.

For the drivers to be comparable, they are measured on a percentage scale from 0 to 100, where 0 represents no implementation and 100 represents full implementation of a driver. By calculating a mean value of the drivers in each of the six CSF categories, the CSFs are measured and can be compared.

The collecting of data for measuring of the drivers can be performed by various methods as interviews, surveys and observations. As a guide when defining the value for a driver, the five-grade scale described in Table 3:1 is used. The guide is not a 5-level Likert scale, where the five levels are the only permissible choices; any value from 0 to 100 is allowed.

Table 3:1 Indicative scale to measure drivers

100	Full implementation
75	Good implementation
50	Neither poor nor good implementation
25	Poor implementation
0	No implementation

The complete set of drivers is presented in Table 3:2.

THEORY

Table 3:2 CSFs and drivers of customer value

1. Delivery Strategy
A. Regular delivery of software; short iterations of equal length
B. Delivering most important features first
2. Agile Software Engineering Techniques
A. Well-defined coding standards up front
B. Pursuing simple design
C. Rigorous refactoring activities
D. Right amount of documentation
E. Correct integration testing
F. Daily builds
G. Code completed at demonstration
3. Team Capability
A. Team members with high competence and expertise
B. Team members with great motivation and self-discipline
C. Managers knowledgeable in agile and having adaptive management style
D. Appropriate technical training to team
E. Appropriate methodology training to team
4. Project management process
A. Following the agile-oriented requirement management process
B. Following the agile-oriented project management process
C. Good progress tracking mechanism
D. Strong communication focus with daily face-to-face meetings
E. Honoring regular working schedule
5. Team environment
A. Co-location of the whole team
B. Coherent, self-organizing teamwork
C. Projects with small teams
D. Projects with no multiple independent teams
6. Customer involvement
A. Good customer relationship
B. Strong customer commitment and presence
C. Customer having full authority.
D. Customers trained in the agile process
E. Use of target-cost contracts to share risk

THEORY

3.5.4 The Drivers and SCS

As described by the SCS in chapter 3.3, projects using Scrum may involve different types of end customers in various ways in the agile process. In some cases the role of the Product Owner is kept in-house, and in others held by the customer. The CSFs and drivers described above define the Product Owner as the customer of a project, not considering whether it is the end customer or not. Therefore the CSFs and total set of drivers are applicable only for end customers being Full Participants according to SCS. For the other types of customers, more or fewer of the drivers are not valid in direct relation to the end customer, and should instead relate to the Product Owner.

Based on the definitions in the SCS, the drivers are in Table 3:3 mapped to the different customer types, and the validity of the drivers for each type is marked out. For Partial Participants, Agile Indirect Participants and Non-agile Indirect Participants some drivers refer to the in-house Product Owner, which are also marked out.

Table 3:3 CSFs, Drivers and SCS customer types

CSFs and Drivers	Customer types			
	Full Participants	Partial Participants	Agile Indirect Participants	Non-agile Indirect Participants
Explanation of symbols: ● Driver is valid ○ Driver is valid with respect to in-house Product Owner – Driver is not valid				
1. Delivery Strategy				
A. Regular delivery of software; short iterations of equal length	●	●	●	○
B. Delivering most important features first	●	●	●	○
2. Agile Software Engineering Techniques				
A. Well-defined coding standards up front	●	●	●	●
B. Pursuing simple design	●	●	●	●
C. Rigorous refactoring activities	●	●	●	●
D. Right amount of documentation	●	●	●	●
E. Correct integration testing	●	●	●	●
F. Daily builds	●	●	●	●
G. Code completed at demonstration	●	●	●	●

THEORY

Table 3:3 CSFs, Drivers and SCS customer types (continued)

3. Team Capability				
A. Team members with high competence and expertise	●	●	●	●
B. Team members with great motivation and self-discipline	●	●	●	●
C. Managers knowledgeable in agile and having adaptive management style	●	●	●	●
D. Appropriate technical training to team	●	●	●	●
E. Appropriate methodology training to team	●	●	●	●
4. Project management process				
A. Following the agile-oriented requirement management process	●	●	●	○
B. Following the agile-oriented project management process	●	●	●	●
C. Good progress tracking mechanism	●	●	●	●
D. Strong communication focus with daily face-to-face meetings	●	●	○	○
E. Honoring regular working schedule	●	●	●	●
5. Team environment				
A. Co-location of the whole team	●	●	●	●
B. Coherent, self-organizing teamwork	●	●	●	●
C. Projects with small teams	●	●	●	●
D. Projects with no multiple independent teams	●	●	●	●
6. Customer involvement				
A. Good customer relationship	●	●	●	○
B. Strong customer commitment and presence	●	○	○	○
C. Customer having full authority.	●	○	○	○
D. Customers trained in the agile process	●	●	○	○
E. Use of target-cost contracts to share risk	●	●	●	-

THEORY

4 THE CASES

This chapter is describing the design of the questions for the studies, the selection of the cases to study, and how to approach the companies. It also presents a short description of each of the studied cases.

4.1 DESIGNING THE CASE STUDY QUESTIONS

This sub-chapter presents areas for data collection and specific investigation questions within each area. The questions are designed to collect data, that when analyzed provide answers to the study propositions. Some of the questions also aim for a wider understanding of the project and its business context, to be used as a foundation from which the more specific questions are analyzed. The areas for data collection described below are *Background*, *Customer characteristics* and *Drivers of customer value*.

For collecting the data an investigation protocol is used, containing all questions. This protocol is described in APPENDIX CASE STUDY DATA. Most data is collected by interviews, and background information is also verified via public information sources such as company websites.

4.1.1 Background

To create understanding for which context a case is analyzed in, and to be able to benchmark, background information about the company, the project and its organization is collected. The questions are designed to provide equal knowledge about all cases and their conditions, as is known for Tactel and stated in chapter 1. The background questions also consider basic facts about the cases needed for application of the theories described in chapter 3, and deal with the following areas:

- General about the company (size, types of products and customers).
- General about the project (size, types of products and customers).
- Which agile method or methods in use.
- For how long the project has been managed with agile methodologies.
- How well the project is following the agile practices “by the book”.
- How the group is organized.
- Experienced and expected benefits with the agile method.

THE CASES

4.1.2 Customer characteristics

This area aims to describe the case's customer types, and to provide data that makes it possible to relate the case and its customers to the SCS. Factors that are investigated are:

- If the project has one or multiple customers.
- If the project has a separate backlog for each customer.

Questions to measure involvement for each customer (end customer, not in-house Product Owner) deal with:

- If the customer is holding the role of the Product Owner.
- If the customer is owning and prioritizing the Product Backlog.
- If the customer is acting as a project member.
- If the customer is collocated with the team.
- If the project, and all its other customers, are transparent to this customer.
- How well the customer knows the agile methodology.
- If the customer is allowed to attend any meeting.
- If the customer is attending the sprint reviews.

4.1.3 Drivers of customer value

The questions in this section are designed to give substance for measurement of the drivers of customer value, and to indicate if the driver is valid for the studied case. The following steps are guiding the investigation of each driver:

1. It is analyzed if the driver is considered in the project, and how the project is working with it.
2. It is analyzed if it is having effect.
3. The degree of implementation is estimated, and a preliminary measure is set.
4. If the degree of implementation is low, it is analyzed if the project is aware of the driver, i.e. if it has been neglected by purpose. And if so, why it has been neglected.
5. A value, 0-100%, is estimated for each driver, based on the interviewees' opinions and the studier's observations and conclusions.

THE CASES

4.2 *PERFORMING THE STUDIES*

By defining criteria for potential cases to meet, it is possible to identify proper case candidates within the same industry context as Tactel. How this is done is described below, followed by a section about the approach for each case study.

4.2.1 **Criteria for the cases**

The enterprises investigated in this study are selected in the same business context as Tactel. This means that they are characterized as high-technology-driven organizations developing software for IT and telecommunications markets. They operate in a changing environment, considering new techniques and fluctuating market demands. Their customers are different types of business customers, and the relation could be described as any (or several) of the following:

- Sub-contracting from major system developers – e.g. outsourced sub-projects for software system vendors.
- New product development – e.g. developing a product for a specific customer.
- Existing products – e.g. selling standardized products to several customers.
- Internal customers – e.g. developing components for a larger system managed within the enterprise.

Implementation of agile methods, preferably Scrum, is also a criterion for the studied cases, and they should not be completely new to agile methods. The studied projects should be able to make competent staff available for interviews, and are allowed to be ongoing or recently completed.

Meeting all criteria qualifies an organization as a case candidate. To increase the response rate, the companies are allowed to be anonymous.

4.2.2 **Selecting the case candidates**

The cases to study are selected to reinforce the SCS and CSF models, by predicting similar applicability on all cases. Studying three cases makes it possible to verify the models iteratively, as shown in Figure 4:1. The initially stated SCS and CSF models (Models I), described in chapter 3.3 and 3.5, are used when collecting and analyzing data in Case A. The result is both an analysis of Case A and an evaluation of Models I. The evaluation results in the

THE CASES

reinforced and adjusted Models II. The same steps are applied to the following case studies, resulting in the final models – Models IV.

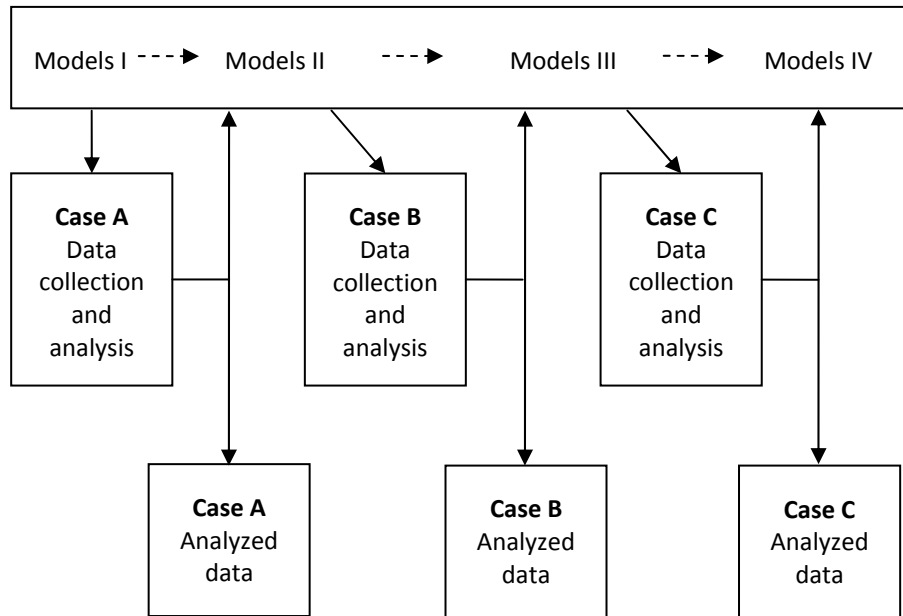


Figure 4:1 Evaluating and developing the models with three case studies

Updating the analysis of Case A and Case B according to the final models, also makes it possible to compare the cases individually, and this study relates the result of Case A with Case B and Case C, as shown in Figure 4:2. This benchmark results in recommendations for Case A to benefit from.

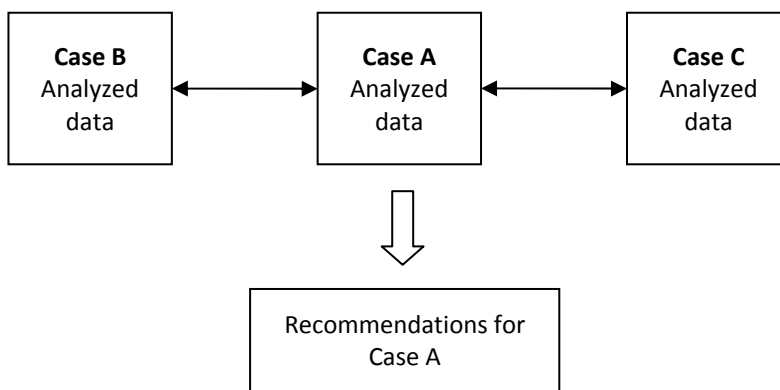


Figure 4:2 Cross-case analysis and benchmark

THE CASES

4.2.3 **Field procedures**

To approach all cases in a similar way, each case study is performed in a step-by-step process:

1. *Initial contact.* The selected company or project is contacted and the case study is presented. It is verified whether the project meets the criteria described above or not, and if it meets the criteria, interviews are booked. If possible they are held with both Scrum Masters and Product Owners, or others with insight in the project and its customer relations.
2. *Preparation.* The main topics to be discussed are e-mailed to the interviewees at the case organization, to prepare them for the interview. The topics are as follows:
 - General facts about the company and the project (organization, number of employees, products, major customers).
 - Customer characteristics and their involvement in the agile process.
 - Experienced and expected benefits with the agile methodology implemented.
 - Engineering techniques (coding standards, simple design, test process, daily builds)
 - Project management and team capability (requirements handling, self-organizing teams, training to team).
3. *Interview.* The interviews are performed according to the investigation protocol in APPENDIX CASE STUDY DATA.
4. *Verification.* The data are grouped and records compiled. Then the preliminary result is gone through with the interviewees.
5. *Analysis.* Each case is analyzed individually based on the verified data, as described in chapter 5 ANALYSIS.

THE CASES

4.3 *THE CASES*

The pilot case – Mobical – was selected to study as this research was defined. Since the research has a qualitative approach, other cases to study were not randomly selected from a large population. Instead a number of potential projects were identified by colleagues within the telecommunications industry and discussed with my supervisor at Tactel. From this population of potential cases, the ones to approach for the study were qualitatively selected considering how they met the criteria, how they could contribute to the reinforcement of the analysis models, and how they could serve as references in the benchmark with Mobical.

Identified were finally a project team at Tactel AB – with different business conditions than Mobical – and a team at the company TAT, The Astonishing Tribe, AB – with somewhat similar business conditions as for Mobical. As they were approached, they agreed to the study.

This sub-chapter provides a presentation of the three cases. Where nothing else is declared, data and statements are taken from the interviews summarized in APPENDIX CASE STUDY DATA.

4.3.1 **Case A – Mobical**

As presented in chapter 1, Tactel AB is a developer of mobile applications, and Mobical – a synchronization and backup service – is one of the developed products. The major customers of Mobical are network operators and service providers worldwide, offering the service to their subscribers. For new customers the system is often in a first step sold as a standardized product with minor customizations, and in later steps new customer-specific features are developed.

The team developing Mobical consists of a Program Manager (PM), a Product Owner (PO), a Customer Project Manager (CPM), and ten developers and testers, of which two also are Scrum Masters (SM). The organization is described in Figure 4:3.

As shown in the figure, Mobical is organized in two Scrum teams – front-end and back-end – with focus on different parts of the system, and with about five engineers (developers and testers) in each.

The Product Owner owns the roadmap for the product, and owns and prioritizes the Product Backlog (PBL). He is also responsible for selling the

THE CASES

product, and has to his support the Customer Project Manager and a seller. The seller assists the Product Owner with sales activities, but has no role in the development process. The Product Owner and the seller have generally the initial contacts with the customers, defining the high-level scope. Then the Customer Project Manager works through the details with the customer, specifying detailed requirements and deadlines, and is their main contact during the project.

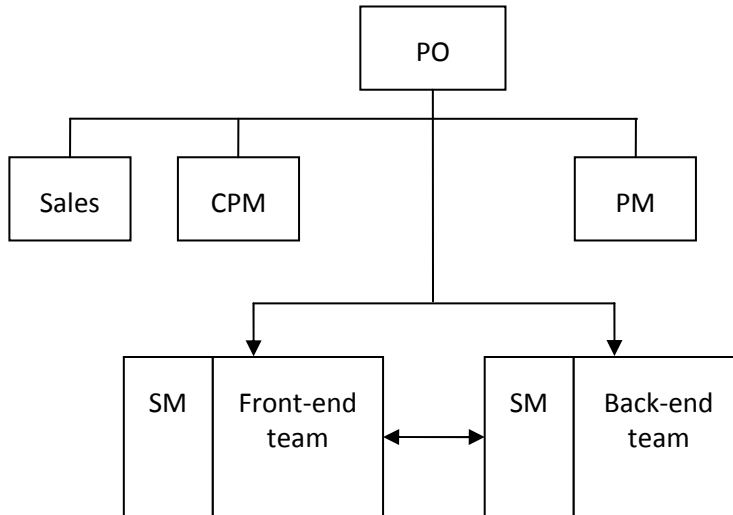


Figure 4:3 Mobical organization from the Scrum perspective

One of the team members also works as Program Manager, which is similar to a line management role. He owns the development, and is among other things responsible for team competence and resource allocation.

The Product Owner is main responsible for adding tasks and prioritizing the Product Backlog, but the Customer Project Manager and the Program Manager edits it too.

To improve the development process, evaluating groups exist for test, development and sales, and two people from each of these groups also form a comprehensive group. All groups meet about every week.

The project has used Scrum for eighteen months, and has evolved Scrum to suit their needs. Before introducing Scrum, there was a lack of structure in the project, and today Scrum is an important project management tool, bringing structure without making any unnecessary planning, which increases flexibility.

THE CASES

The visibility is also increased; the prioritizations from sales and customer responsible are visible for everybody in the project, and the awareness of what to do and what everybody is working on is higher. It is also easier for the Product Owner to plan and negotiate with customers about delivery dates. With Scrum, the development process has better abilities to improve, and it constantly is; as an example the process for handling defects is mentioned. Though, the project is still complex and hectic since tasks are added during sprints and several customers with different delivery dates and priorities are handled simultaneously.

Decreased time to market, higher quality and decreased overtime are not significant benefits for Mobical. Estimation of the time needed to complete tasks may have improved, but since the use of metrics is poor this is uncertain.

Future potential benefits can generally be identified in all of the areas development, test, customer project management and customer collaboration. A more extensive use of relevant metrics could increase visibility for the team and others, and also increase engagement and commitment by the team.

Mobical considerations

At Mobical four interviews were held; with the both Scrum Masters, of which one is also working as tester and one as developer, with the Product Owner and with the Program Manager.

The Program Manager, who until recently was the Scrum Master, is having very good knowledge in Scrum and agile methods. He has extensive knowledge in all Mobical's sub-systems and knows the customers. Therefore his opinions generally are expected to be more valid than the junior Scrum Masters'. He might, however, be biased by being the one introducing Scrum to the project, and by being responsible as Scrum Master and Program Manager for a long time.

The new Scrum Masters are not as experienced in the whole system as the Program Manager, but may bring light to the studied areas from test and development perspectives. Their views therefore complement the Program Manager's, but may also bias their responses.

The Product Owner brings – as responsible for customers, requirements and the Product Backlog – expertise knowledge to customer-related areas. Since he is not working in the team, he is not expected to have full knowledge about engineering techniques and other team-internal processes.

THE CASES

From area to area the different opinions were weighted and bias tried to be detected. When the collected data was put together, the primary analysis was shown to the interviewees to get a second opinion if the right views were considered and biased opinions filtered. Their second opinions are considered in the presented material.

4.3.2 Case B – Audio Control

The second case – Audio Control – is also a project at Tactel AB, which develops and maintains a software module in Sony Ericsson’s mobile phones. Audio Control is run as a sub-project in Sony Ericsson’s organization, and is contracted on an outsourced basis. This means that only one customer exists, which is buying development hours rather than a product; a group of engineers during a specified period of time. The development consists of about 90% maintenance and support and 10% new development. Tactel has handled Audio Control as an outsourced project for more than five years.

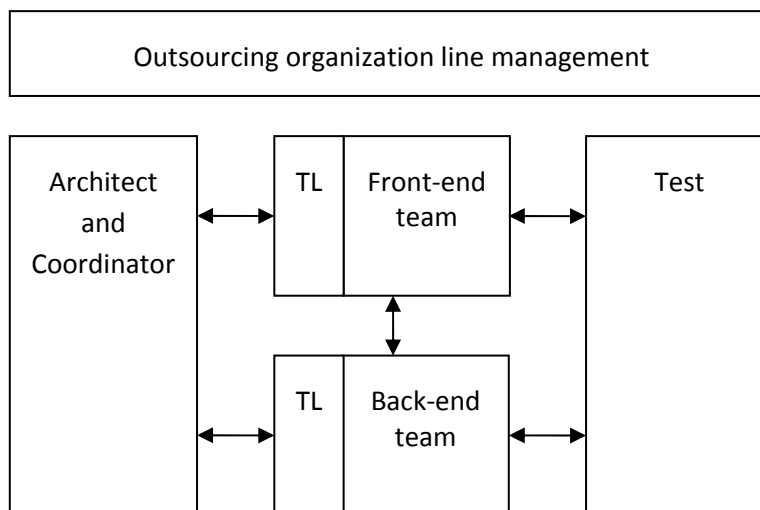


Figure 4:4 Audio Control organization

The organization handling Audio Control within Tactel is described in Figure 4:4. Business relations between Tactel and the customer are handled by line management, and consider several other projects besides Audio Control. Development is managed in two separate Scrum teams – Back-end and Front-end – who are supported by an architect and a technical coordinator outside the Scrum process. All software is verified by an internal test department,

THE CASES

collocated with the teams. The testers support both development teams, and work outside the agile process.

This study focuses on the front-end team consisting of five developers, of which one is the Team Leader (TL). He is having the role of Scrum Master, and also has main contact with the customer.

The customer sorts and prioritizes all requirements in a task-management tool, which works as a product backlog (and is hereafter referred to as the PBL). At the sprint planning, the Team Leader is representing the customer, and the role of the Product Owner could be said to be divided between the customer and the Team Leader.

The project is managed with Scrum for three months. Since short sprint-lengths are practiced (two weeks) the team members no longer see themselves as beginners to the method, and they have managed to define their own version of Scrum. For those reasons the project is considered to meet the criteria, and the different type of customer and product makes Case B a good complement to Case A in the analysis and reinforcement of theories.

Audio Control experiences that the communication climate has improved with Scrum; everyone knows what the others are working on, they are more interested in each other's work and get faster help and answers to their questions. They feel that the short sprints make it possible to start from a blank sheet every other week; to evaluate the past and make a new planning. Also prioritizations and structure in the project are clearer, making the work process more efficient, and today there is time to build code with higher quality.

The efficiency could probably be increased even more with better metrics and follow-ups of them, but at the same time risking that time spent on reporting would increase. The project also sees possible future benefits if the customer adapts its processes to Scrum. Then prioritizations could be clearer and releases better aligned with the sprints.

Audio Control considerations

At Audio Control interviews are held with the Team Leader and with one of the developers. They both have very good knowledge in the development process and good knowledge in Scrum. The Team Leader, handling almost all contacts with the customer, is experienced to have superior knowledge about the

THE CASES

customer relations, and within this area his opinions are considered to be more valid.

Since the project is quite new to Scrum, they might be over-enthusiastic and therefore biased in their responses. Though, this risk is considered to be small since the project for several years is run very well and delivering good results.

Within the different areas opinions from the developer and the Team Leader were weighted and bias was tried to be detected. When the collected data was put together, the primary analysis was shown to the interviewees to get a second opinion if the right views were considered and biased opinions filtered. Their second opinions are considered in the presented material.

4.3.3 Case C – Cascades and Kastor

TAT, The Astonishing Tribe, AB (TAT) is a Swedish software technology and design company offering products and services that differentiate and enhance the user experience of portable devices (21). TAT's mission declares that "TAT is a company where design and technology are two sides of the same coin. It requires a culture and a company philosophy that creates a working environment that encourages change, curiosity and exploration."

TAT offers a suite of products and services for the creation of advanced mobile user interfaces. The TAT market offering is divided into two main areas, products and services, and the main offerings are as follows.

- *TAT Cascades* is a framework for the production of advanced user interfaces.
- *TAT Kastor* is a powerful rendering platform.
- *TAT Motion Lab* is an XML development environment for TAT Cascades.
- TAT also offers complete UI design projects.

The company is headquartered in Malmö, Sweden, and has local offices in Korea and USA. TAT works with 4 of the 6 leading Original Equipment Manufacturers in the mobile device space today. Publicly announced clients include Sony Ericsson, Motorola, S60, Samsung, Vodafone and Orange. (21)

Since the company was founded in 2002 it has continuously grown, and employs in November 2008 about 150 people. Agile and iterative development methods are used in several projects within TAT, but all are not working specifically according to Scrum.

THE CASES

The studied case is the development of Cascades and Kastor. The development of both products is much related since the main product Cascades is built on Kastor, they have the same Product Owner, and they have about the same customers. Therefore they are in this study considered as one project, where different Scrum teams work with the different products.

Scrum has been implemented for about two years, and the development is organized in two Scrum teams – Cascades and Kastor – with about ten respectively six developers. Beside the teams work a Product Owner and two Scrum Masters (one for each team). Function and feature tests are performed at a separate test department at TAT, not involved in the Scrum process. Other product development teams also work by Scrum and close to Cascades and Kastor, with which dependencies also exist. They are referred to as Product Team B and C in the visualization of the organization in Figure 4:5.

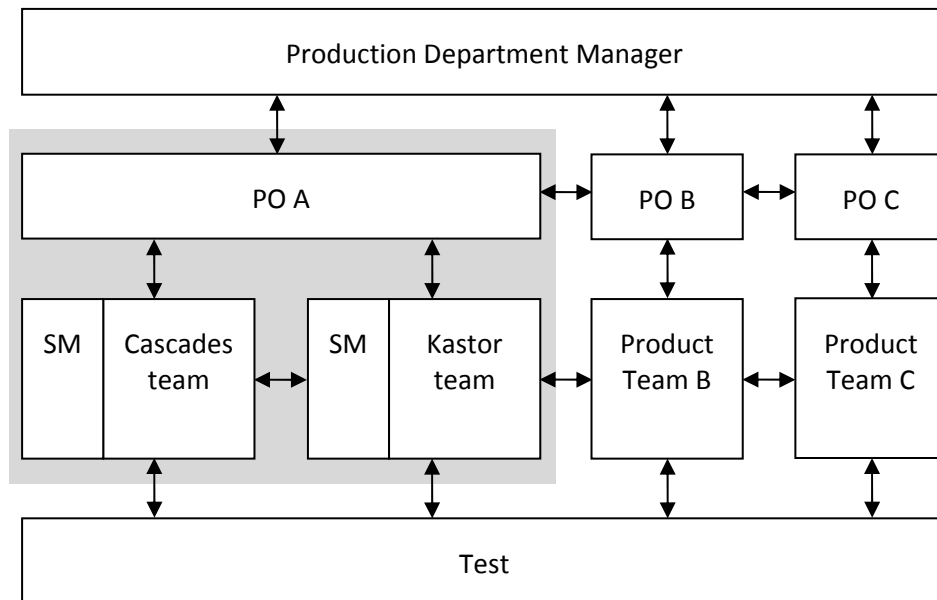


Figure 4:5 Cascades and Kastor organization

Temporary *Feature Teams* are on occasion put together when the development of an important task or feature concerns two or more of the products and corresponding teams. The Feature Teams then contain members from the concerned product teams.

THE CASES

The Product Owner is also part of the sales and marketing organization together with a Key Account Manager (KAM) and a Technical Account Manager (TAM). As described in Figure 4:6, the Product Owner is responsible for the road maps of the products, and builds and prioritizes the PBLs based on them. The organization is about to put together the different PBLs to one single, to align the development of all teams and better handle dependencies.

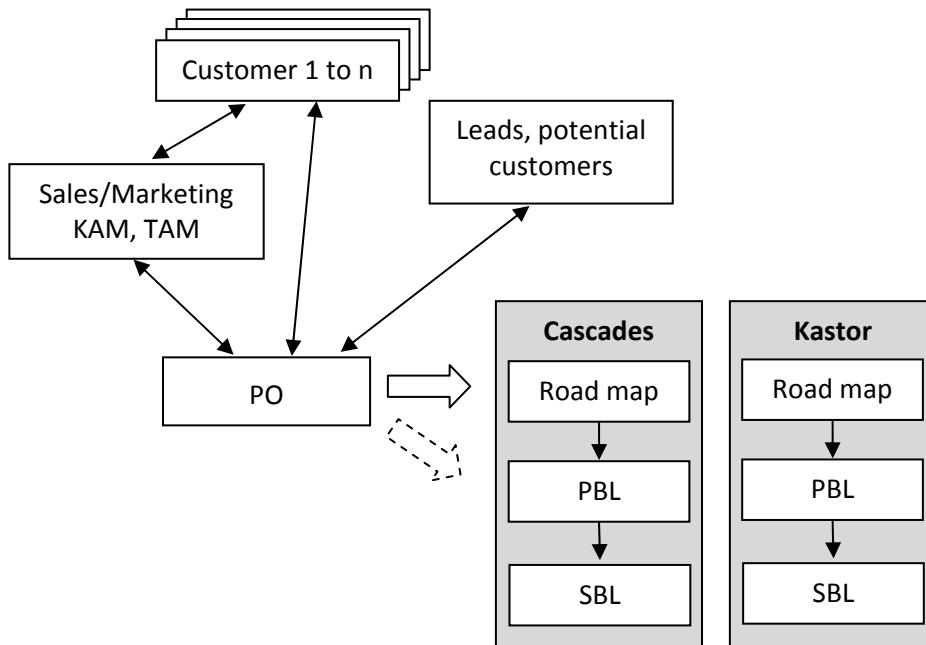


Figure 4:6 Cascades and Kastor sales and marketing organization

Cascades and Kastor are sold as products by license, with a new software release about every three months. The project has about five key customers, five additional major customers and some minor ones. The key customers are mainly influencing the development by putting requirements on future releases. The additional major customers are also requiring features to develop and bugs to fix, but are not prioritized as high. The minor customers' requirements have lowest priority to include in future releases. Some customers are public announced, but peer priorities are not transparent to anyone outside the project.

Further requirements and features are added to meet requirements from potential future customers. This is done by having a dialogue with them, by being updated in new technology and by following the *leads* – the leading developers – in the business. This is also described in Figure 4:6.

THE CASES

The project experiences that the introduction of Scrum brought a common vocabulary to the project, which have resulted in a better structure. This helps the teams to focus on the right things, supports regular follow-ups and re-prioritizations, and enables continual risk management. Communication is also improved, leading to better problem solving.

Time to market is generally not reduced by Scrum, but the products are easier adapted to the end customers' needs, and this improved customer-orientation has resulted in new business.

The process is continuously evaluated, and potential improvements are identified within better coordination and co-location of teams, automatic tests, use of metrics, defining criteria for when a task is done, and generally reduce waste. Further improvements could focus on increased development speed and quality, but may not necessarily be enabled only because of Scrum.

Cascades and Kastor considerations

At TAT interviews were held with the Product Owner and with the Scrum Master for the Cascades team. They are both very familiar with the development process, and the Product Owner also has good knowledge of the customers. Therefore the Product Owner's opinions might be considered more important in questions about the customers. The Scrum Master also has the role of super Scrum Master, with responsibilities for the agile processes within all development teams. He also works closer to the teams, and his opinions about the teams and development routines and practices are generally considered more valid.

Since the time for the investigations was limited, personal relations could not be established with the interviewees. Therefore they might, consciously or unconsciously, have been a bit careful about what information to share, to not risk revealing any business secrets. There may also be a risk that the interviewees see the investigator as an external part, and they might tend to speak more freely about things that work well in the project and tell less about things that do not. This kind of bias is hard to neutralize, and it might shift the values in the analysis of the specific case and in the benchmark.

Though, the different opinions are weighted and from area to area bias is tried to be detected. When the collected data was put together, the primary analysis was shown to the interviewees to get a second opinion if the right views were considered and biased opinions filtered. Their second opinions are considered in the presented material.

5 ANALYSIS

This chapter analyzes the collected data, related to the theory and the research questions. First, the compiled and verified records from the case studies are analyzed in relation to the theories and models, as initially described in chapter 4.2.2 Selecting the case candidates. For each case the applicability of the SCS and CSFs are analyzed, and the models are adjusted and reinforced. The analysis of the models is then summarized, and later used to discuss the propositions “Which drivers of customer value can be identified in agile practices?” and “How can customers be classified in matter of participation in an agile project?”

At the end of the chapter Case A is benchmarked with the other cases, and cross-case conclusions are drawn to analyze how aware the studied projects are of the benefits with agile methods. This section’s findings are later used in the discussion of the proposition “How well are the drivers and customer types considered by companies in Tactel’s business context?”

The fourth proposition “How can Mobical benefit from the findings?” is discussed in chapter 6 in light of both the analysis of the single Mobical case and the benchmark.

The data analyzed in this chapter is presented in APPENDIX CASE STUDY DATA.

5.1 CASE A – MOBICAL

5.1.1 Case A – Model reinforcement 1

As described below in chapter 5.1.2, the SCS clearly shows the position of Mobical’s customers, and it is possible to collect and analyze relevant data in an unambiguous way. Therefore the SCS does not need any adjustments, and is reinforced by being totally applicable on Mobical.

The CSF analysis described below in chapter 5.1.3 shows that some of the drivers defining the CSFs are hard to analyze and measure in explicit ways. They may be too alike or do not have relevance as separate drivers. By putting drivers within a CSF together, focus is moved from unique practices to the

ANALYSIS

general implementation of an area. This makes the data collection and analysis more explicit, and minimizes the risk of one single driver incorrectly shifting the value of the CSF.

The changes in the CSF theory are made within the area of Agile Software Engineering Techniques, and presented in the following paragraphs. For the other CSFs the data collection and analysis is experienced to be unambiguous, hence strengthening them.

The drivers *2.B Pursuing simple design* and *2.C Rigorous refactoring activities* are hard to analyze each at a time, since they describe the same phenomenon. Refactoring activities is an underlying practice when working with simple design, and should therefore rather be considered as a part of driver 2.B than being a driver of its own. Consequently driver 2.C is removed.

Driver *2.D Right amount of documentation* is somewhat confusing in its word choice, since it is hard to measure what is the *right amount*. What really is relevant to analyze is how a project works with documentation – requirements, backlogs, design documents, test specifications etc. – and the agile measures in the documentation process. As a result the driver is re-phrased to *Agile documentation handling*.

Driver *2.E Correct integration testing* measures the implementation of test through the whole process; the use of unit tests, automatic tests and function tests. Therefore the driver is more explicit when re-phrased to *Agile test process*.

These adjustments of the theory result in the updated set of drivers of *Agile Software Engineering Techniques* as shown in Table 5:1.

Table 5:1 Reinforcement of Agile Software Engineering Techniques

2. Agile Software Engineering Techniques
A. Well-defined coding standards up front
B. Pursuing simple design
C. Agile documentation handling
D. Agile test process
E. Daily builds
F. Code completed at demonstration

ANALYSIS

5.1.2 Customer classification

Mobical has a very similar relation to all of its customers, making it logical to classify them as of the same type. Though, differences in importance exist, but not in the way they are approached.

The customers cannot be said to be direct participant according to the SCS since they are not holding the role of the Product Owner or are allowed to attend any meetings within the agile process. They are clearly indirect participant, and the transparency for a customer is minimal, placing the customers in the lowest part of the SCS matrix.

Mobical's products are bought with customer-specific adaptations, refining a standardized product, and the in-house Product Owner is responsible for providing market value in the product and prioritizing adaptations to specific customers. This makes the customers Non-agile Indirect Participants according to the SCS. Though, there are almost always a dialogue between the Customer Project Manager and the customer, and changes in the requirements are made during the projects. Deliveries are also tried to be aligned with the sprints, placing the customers somewhat to the right in the Non-agile-Indirect-Participants box, as shown in Figure 5:1.

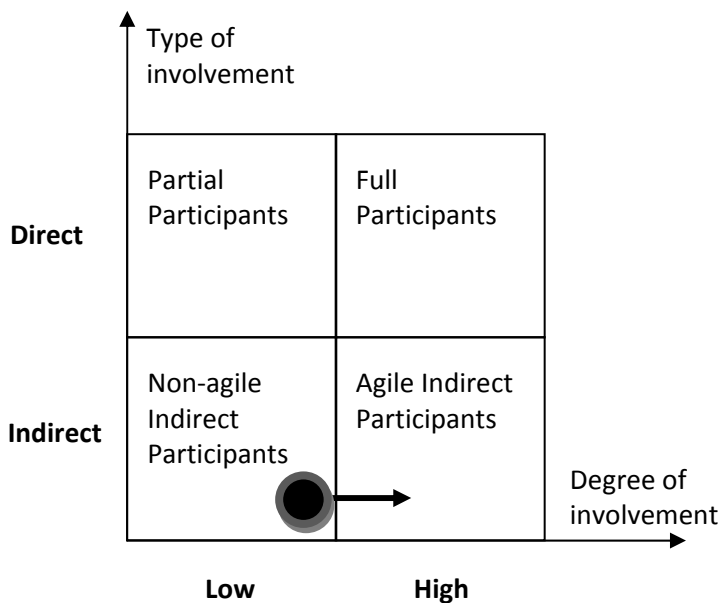


Figure 5:1 Position of Mobical's customers in the SCS

ANALYSIS

It might be possible for Mobical to align customers more with the Scrum process, especially when adding features to a system at an existing customer. This means that a transition against the Agile-Indirect-Participants box could be beneficial for mature customers, which is also indicated in Figure 5:1. Though, the general approach seems to suit the current customers' wants and needs well.

5.1.3 CSFs and Drivers

Since the customers are classified as Non-agile Indirect Participants, the Product Owner is considered to be the customer representative in the Scrum process. In Mobical's case this role is somewhat divided between the Product Owner, the Customer Project Manager and the Program Manager. Therefore all of them have to be considered when measuring drivers related to the customer.

The results of the driver investigation, for which the data is described in APPENDIX CASE STUDY DATA, are presented per CSF in Figure 5:2 and per driver in Figure 5:3.

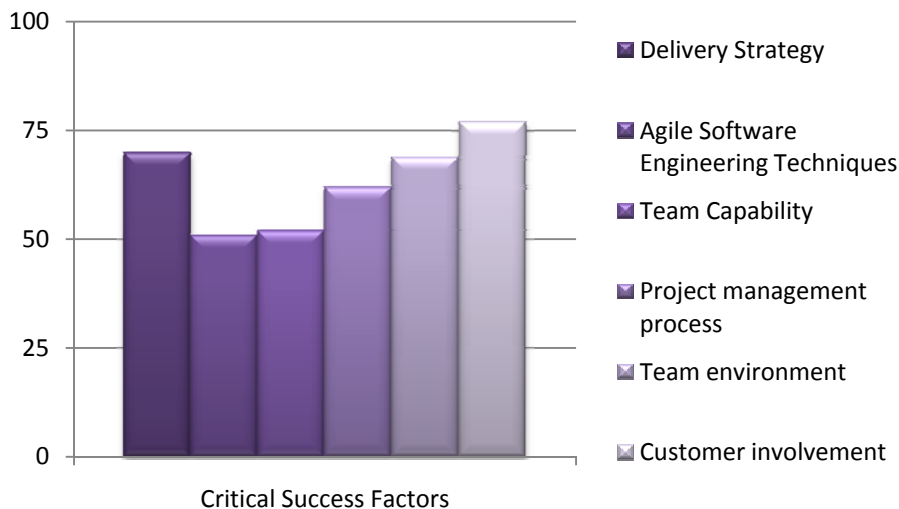


Figure 5:2 Implementation of the CSFs in Mobical

Figure 5:2 shows that the CSFs generally are implemented between *neither good or bad* and *good*. The CSF bars are drawn in decreasing order with the most critical – *Delivery Strategy* – to the left, and the diagram points out that the second and third most important factors have the lowest degree of

ANALYSIS

implementation. In the subsequent paragraphs the CSFs are analyzed individually.

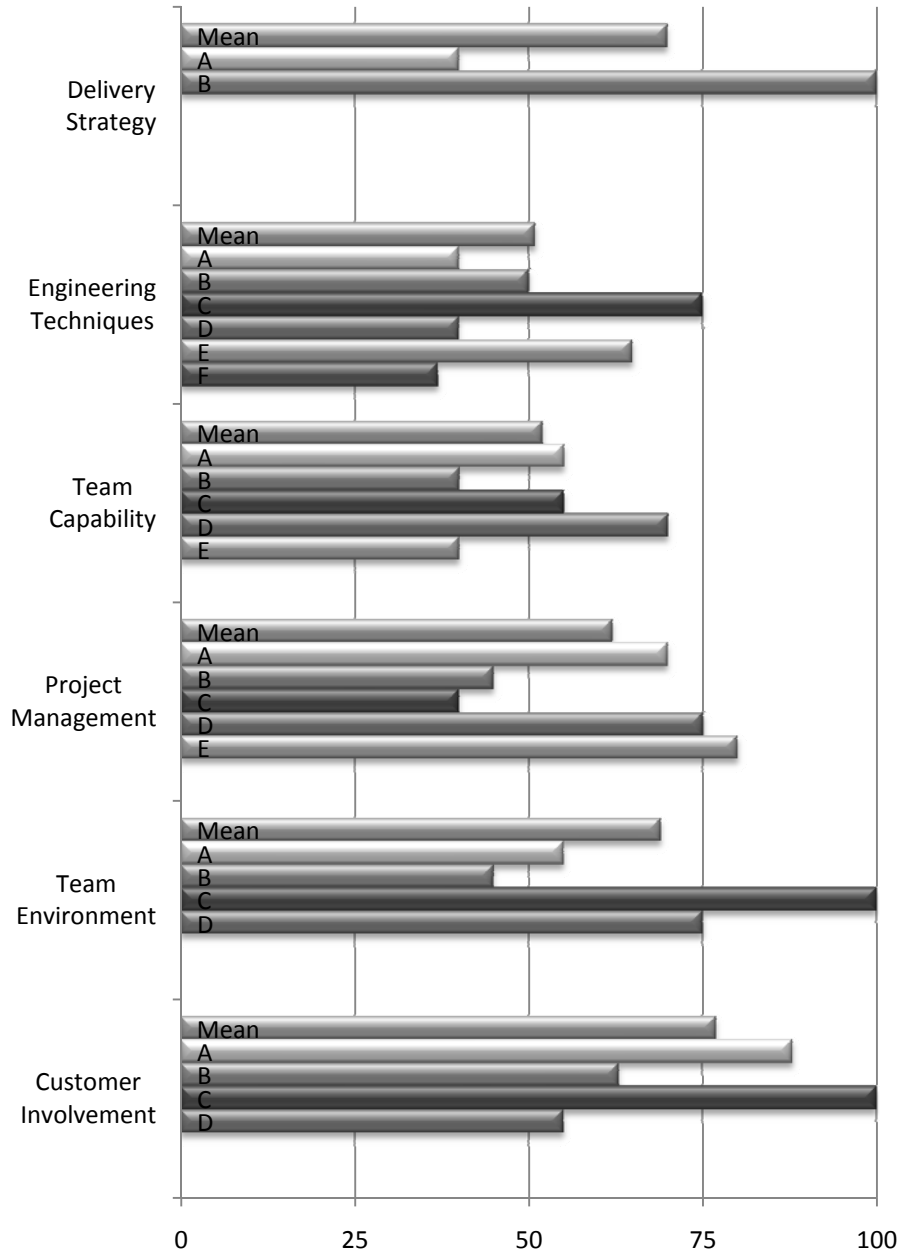


Figure 5:3 Implementation of each driver in Mobical

ANALYSIS

The delivery strategy, visualized in Figure 5:4, is generally implemented quite well, primarily because the most important features always are developed and delivered first. Even if the sprint lengths are two weeks, the project fails to finish all tasks within a sprint, and seldom delivers increments of shippable functionality. Criteria for when a task is complete do not exist, and features are seldom demonstrated on the target systems.

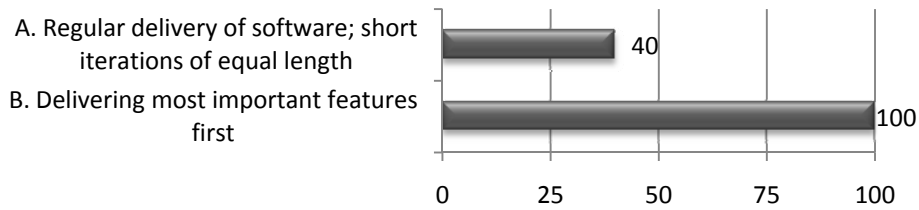


Figure 5:4 Mobical Drivers of Delivery Strategy

Agile Software Engineering Techniques is totally the CSF with the lowest scores. What is implemented in a good way is the documentation handling and the use of a Wiki, but less attention is paid to coding standards, automatic tests, and keeping the code clean. The code could hardly be complete at demonstrations as no criteria for when a task is complete exist, and this driver has the lowest degree of implementation. The drivers of Agile Software Engineering Techniques are shown in Figure 5:5.

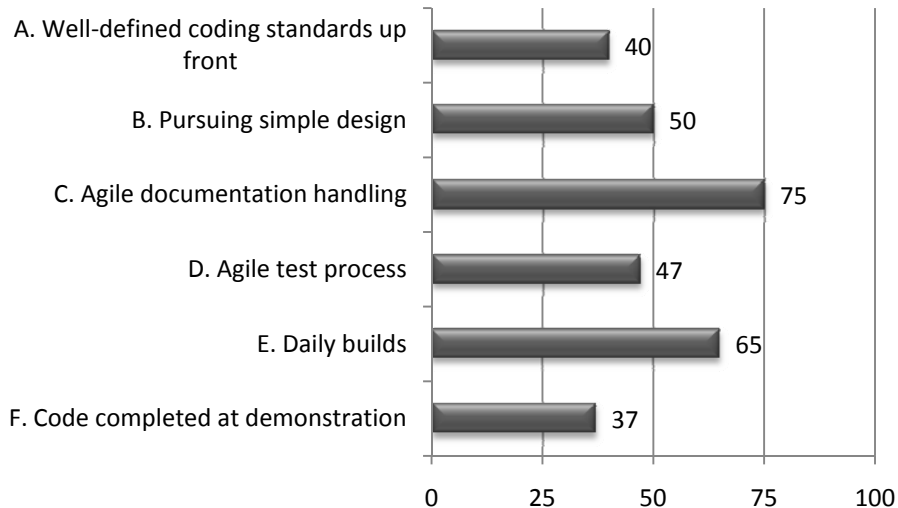


Figure 5:5 Mobical Drivers of Agile Software Engineering Techniques

ANALYSIS

The teams' technical capabilities are rather good, and the engineers are quite senior, but the teams are not totally cross functional. Some team members have unique expertise knowledge (test, database etc.) and sometimes become bottlenecks. This also makes many team members care of "their own" tasks, not taking common responsibility for all Sprint Backlog (SBL) items.

Since there is also a common understanding in the project that there is nothing wrong with not fulfilling all tasks within a sprint and measures of progress are poor, all team members are not very motivated to work agile. The methodology knowledge is quite good for the team members in general, but enough time is not invested to improve and learn.

The Product Owner and the Program Manager are collocated, synched individually, know Scrum well and try to synch deliveries with sprints. The Customer Project Manager, however, is not fully synched with Scrum and sprints, and is not collocated with the others. The poor synch of deliveries and sprints could depend on the Customer Project Manager being responsible for "old" customers with habitual patterns, but also because Fridays (sprint-delivery days) are not desired for deliveries.

The measure of each driver for Team Capability is presented in Figure 5:6.

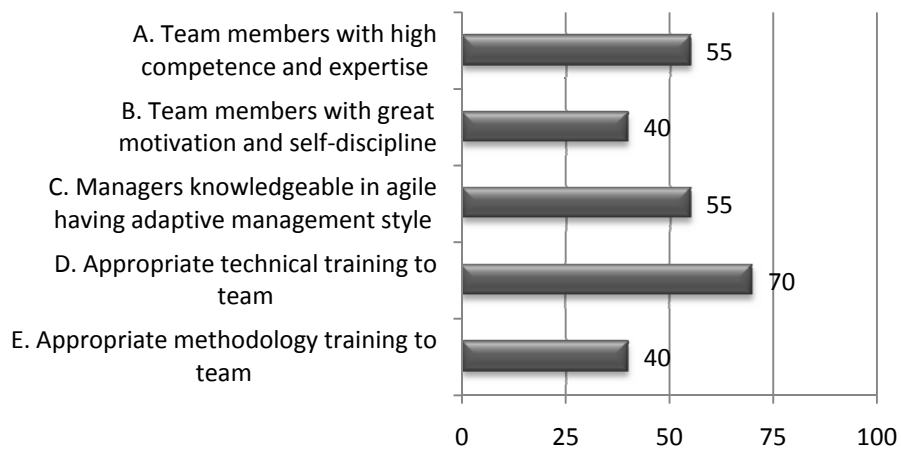


Figure 5:6 Mobical Drivers of Team Capability

As shown in Figure 5:7, the requirement management process works quite well. The requirements are, however, written in traditional documents, and translation to the backlogs is not optimized. Many tasks are added during the

ANALYSIS

sprints, and to reserve time in the sprint planning placeholder-tasks are used. Added tasks are equivalent to about 50% of the time in a sprint, making estimation very hard, and as a consequence all tasks are not finished within the sprint. This could be solved by working overtime, but today this option is only used in shorter periods at customer deliveries.

Self-organization of the teams works quite well at the sprint planning, but they tend to not being able to self-reorganize when tasks are added during the sprints. The communication climate is good, and except the Customer Project Manager everybody attend the daily face-to-face meetings. In spite of this, the team members do not take common responsibility for the tasks, and an “end-of-sprint spirit” does not exist.

The project management tool used is not bad, neither optimal, and the ones that need to see progress understand the tool, but outside stakeholders probably do not. The follow up of metrics – velocity (finished work within a sprint), feature turnaround speed (from story in PBL to on the market) etc. – is poor, and team members are not always motivated to be disciplined in reporting.

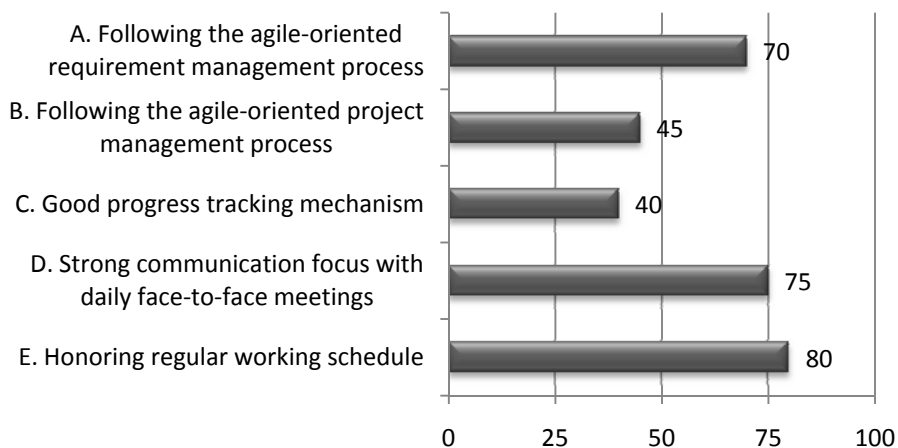


Figure 5:7 Mobical Drivers of Project management process

The two development teams work with different parts of the product (front-end and back-end) to minimize the dependencies between the teams, and as visualized in Figure 5:8 the sizes of the teams are small. Though, the self-organization of the teams is not optimal. Both teams are located in the same room, emphasizing a good communication environment, but the fact that the

ANALYSIS

Customer Project Manager is located in another city results in poor communication between him and the team.

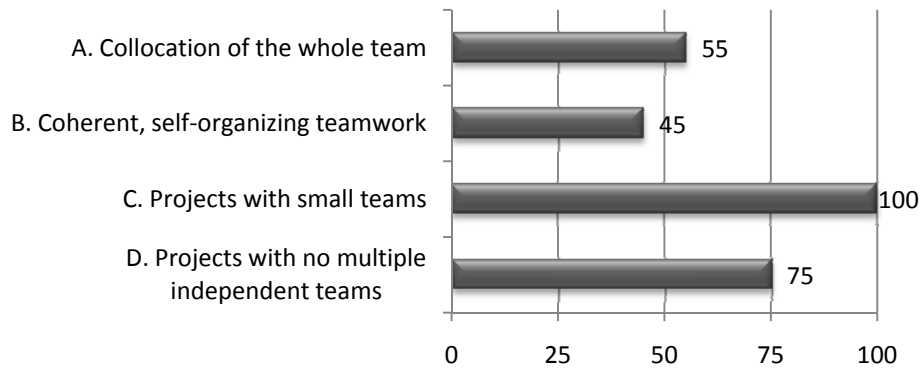


Figure 5:8 Mobical Drivers of Team environment

As shown in Figure 5:9 the involvement of the customer, i.e. Product Owner, Customer Project Manager and Program Manager, is high. What lower the measures are the Customer Project Manager not being that present and not equal to the team in knowledge of the agile process.

The target-cost approach is hard to apply when selling and delivering a standardized product. Target-cost contracts are also not applicable for Mobical's recent customers, since they are very price-orientated and anxious to keep their budget. Therefore this driver has been left out of the analysis.

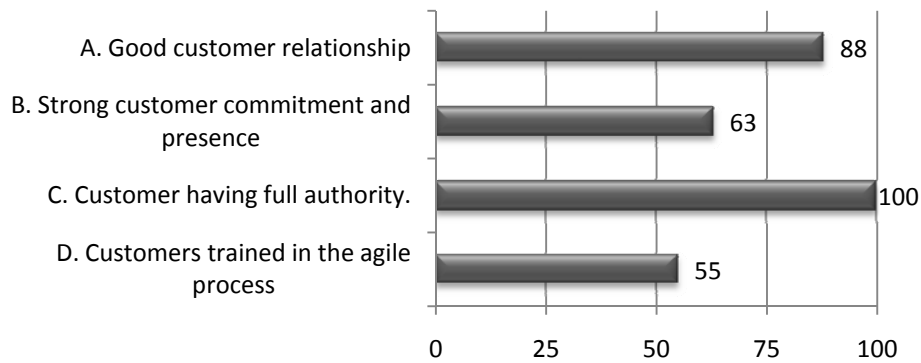


Figure 5:9 Mobical Drivers of Customer involvement

ANALYSIS

The drivers that are least considered and have the lowest degree of implementation (a score under or equal to 50) are presented in Figure 5:10.

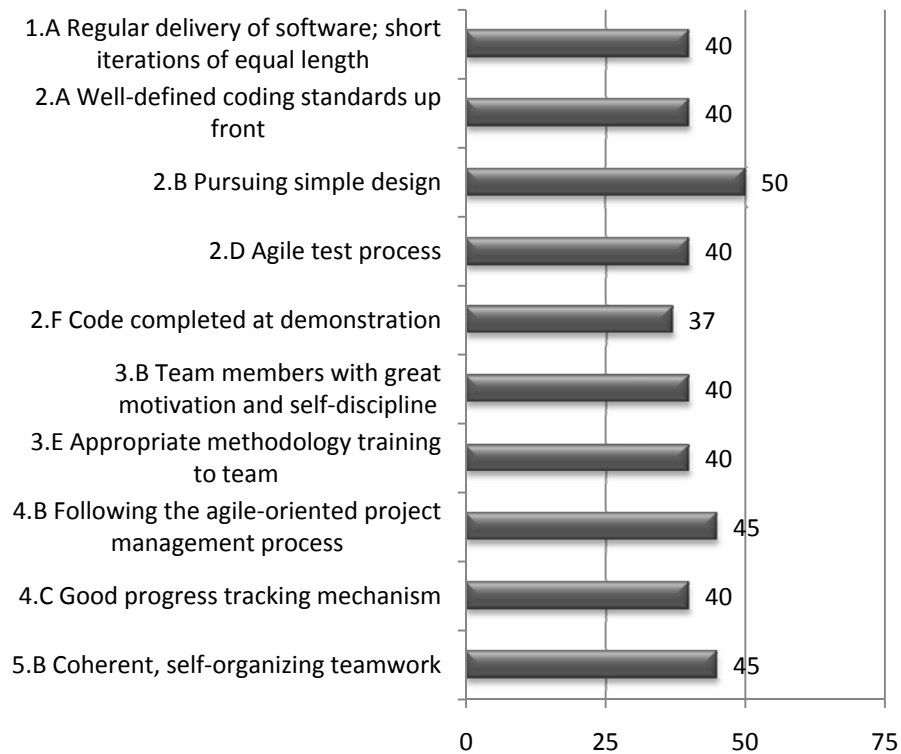


Figure 5:10 Least considered drivers in Mobical

The factors most significant, underlying the drivers with low scores, are:

- the absence of criteria for when a task is done
- the adding of too many tasks during the sprints
- team members' shortcomings in motivation, ability to self-organize and common responsibility
- the off-location of the Customer Project Manager
- the lack of metrics making it possible to follow up progress
- the undefined engineering process (coding standards, test routines etc.)

By considering and improving these factors, many of the drivers will be better put into practice, the development process improved, and a better product developed. In this way both the internal working environment and the perceived end-customer value will increase.

ANALYSIS

5.2 CASE B – AUDIO CONTROL

5.2.1 Case B – Model reinforcement 2

The SCS, as described in chapter 5.2.2 below, clearly shows the position of Audio Control's customer, and it is possible to collect and analyze relevant data in an unambiguous way. Therefore the SCS does not need any adjustments, and is strengthened by being totally applicable on the case.

Nor needs the CSF model any adjustments to make data collection and analysis explicit, and the case reinforces this theory too. The CSF analysis is presented below in chapter 5.2.3.

5.2.2 Customer classification

Audio Control has a single customer, but 3-4 different departments in the customer's organization are receivers of deliveries – and could then be seen as different customers. Though, they sort and prioritize their requirements in the same PBL, where all tasks from the different departments are prioritized by the customer. The project also works according to the customer's standards, and hence this study identifies Audio Control to have one single customer classified as Direct Participant, placed at the top of the SCS matrix.

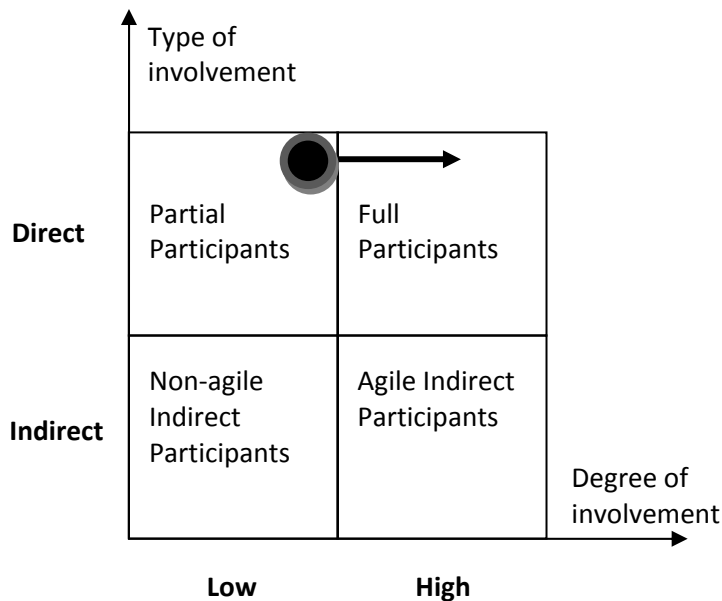


Figure 5:11 Position of Audio Control's customer in the SCS

ANALYSIS

The customer is not entirely acting as a project member, is not collocated with the team and does not possess a complete understanding of Scrum. However, the transparency of the project is very high, and the role of the Product Owner is divided between the customer and the Team Leader. This classifies the customer as Partial Participant, placed to the right in the Partial-Participants box in Figure 5:11.

If the customer aligns its processes better with Scrum and takes full responsibility as Product Owner by attending sprint planning meetings etc, structure and prioritizations could get even clearer in the project. This possible evolution would mean a transition of the customer to the Full-Participants box, also indicated in Figure 5:11.

5.2.3 CSFs and Drivers

Since the customer is classified as Partial Participant, and the role of the Product Owner is divided between the customer and the Team Leader, the latter is considered as *customer* when measuring the drivers 6.B and 6.C. For all other drivers the end customer is considered, as described in Table 3:3.

The results of the driver investigation, for which the collected data is described in APPENDIX CASE STUDY DATA, are presented per CSF in Figure 5:12 and per driver in Figure 5:13.

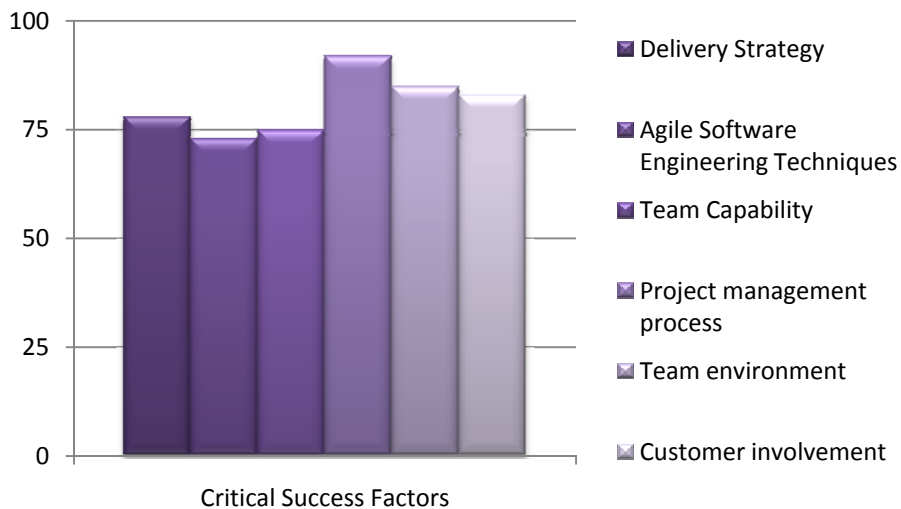


Figure 5:12 Implementation of the CSFs in Audio Control

ANALYSIS

Figure 5:12 shows that the CSFs generally are well or very well implemented. This corresponds in a good way to the overall impression of the project being very healthy and delivering good results. The chart points out that the three most critical factors have the lowest degree of implementation, and that the second most important factor has the very lowest – though, with high scores.

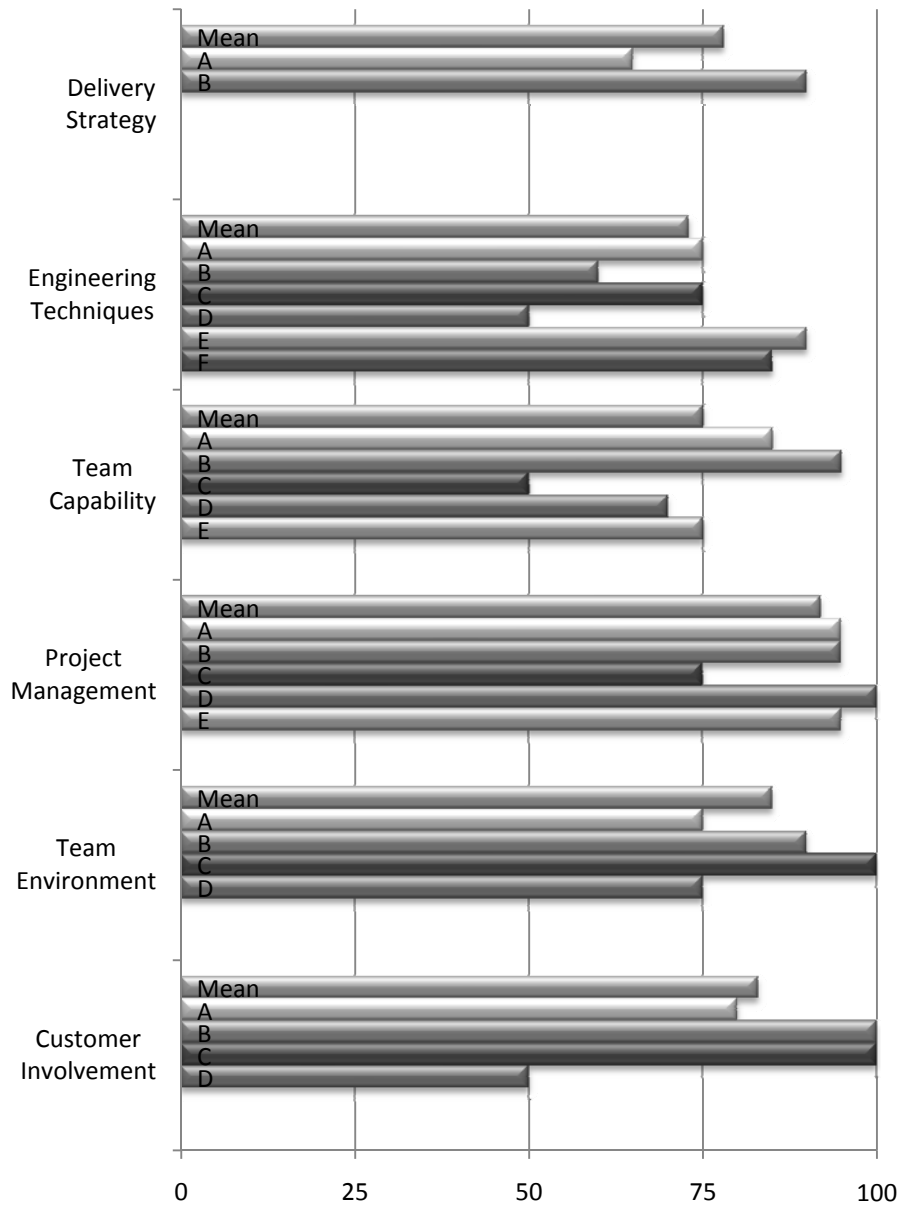


Figure 5:13 Implementation of each driver in Audio Control

ANALYSIS

The delivery strategy, visualized in Figure 5:14, is generally well implemented, and the most important features are nearly always developed and delivered first. Sprint-lengths are two weeks, but deliveries are not aligned with them. Instead support and maintenance tasks are delivered as soon as they are finished, as this is requested by the customer. New developed features could be better aligned with the sprints.

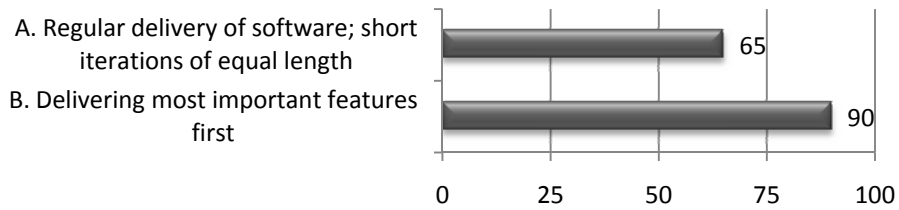


Figure 5:14 Audio Control Drivers of Delivery Strategy

Agile Software Engineering Techniques, shown in Figure 5:15, is totally the CSF with the lowest scores. The practices implemented in a good way are regular builds and the definition of when a task is complete. Future improvements can be made within simple design and test-driven development, and implementations of these areas are a bit lower.

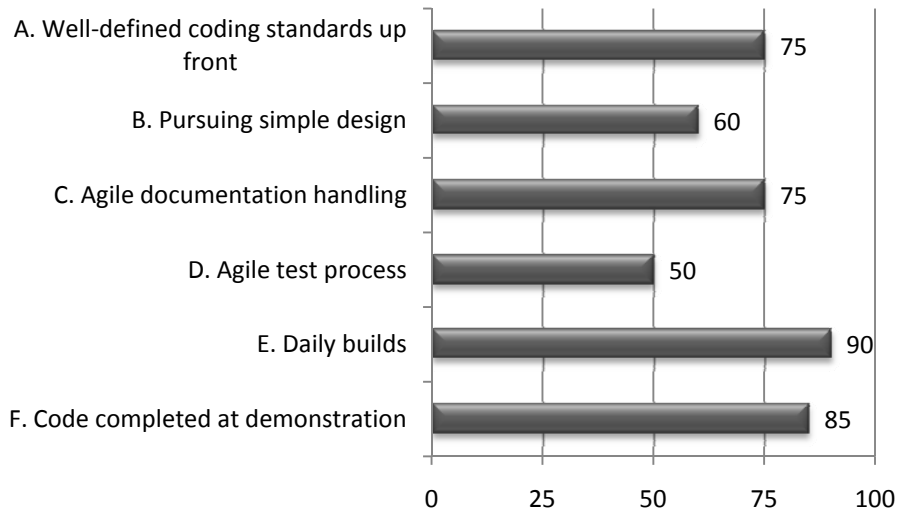


Figure 5:15 Audio Control Drivers of Agile Software Engineering Techniques

ANALYSIS

The team's capabilities are very good; the engineers are rather senior, the team is quite cross functional and all members are motivated. Some team members are more experienced than others in certain areas. However, this is not hindering the development, and tasks within the expertise areas are divided between all developers. This also gains a common responsibility for the work.

Since Scrum only has been implemented for a short period of time, it is hard to measure the level of methodology training. This far all team members have been educated and know Scrum well enough and there has not yet been a need for strategies for future methodology improvement. The lack of strategies, however, keeps the score down to *good*.

The managers within the customer organization respond positively to Scrum, but do not really know Scrum and are not yet taking part of the process themselves, e.g. by adapting reporting structures and deliveries to Scrum.

The measure of each driver for Team Capability is presented in Figure 5:16.



Figure 5:16 Audio Control Drivers of Team Capability

ANALYSIS

As shown in Figure 5:17, nearly all parts of the project management process works very well. The tasks are easily transferred from the PBL to the SBL, the team meets every day and deals with the tasks, and the process seems to run very smooth. The team seldom works overtime, which also was the situation before the introduction of Scrum.

A web-based Wiki is used for handling the SBL. This is a very simple tool, which was easy to implement and that is simple to report in. The visibility of the progress is good enough for the project, but there is poor support for metrics, e.g. burn-down charts and velocity. Though, the good communication climate makes progress clear for at least the Team Leader, and there is no critical need for a more detailed project management tool.

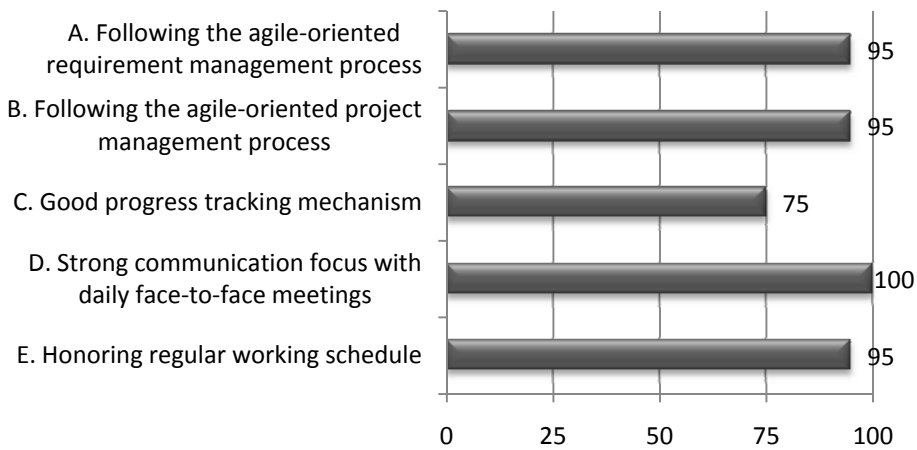


Figure 5:17 Audio Control Drivers of Project management process

Figure 5:18 visualizes the drivers of Team environment. Audio Control is divided in two small development teams responsible for different areas of the common code base (front-end and back-end). Though, dependencies exist, and despite the teams being located in the same office space, communication between them is poor.

The Team Leader is collocated with the team, but slightly distanced from the others, which somewhat affects the communication.

ANALYSIS

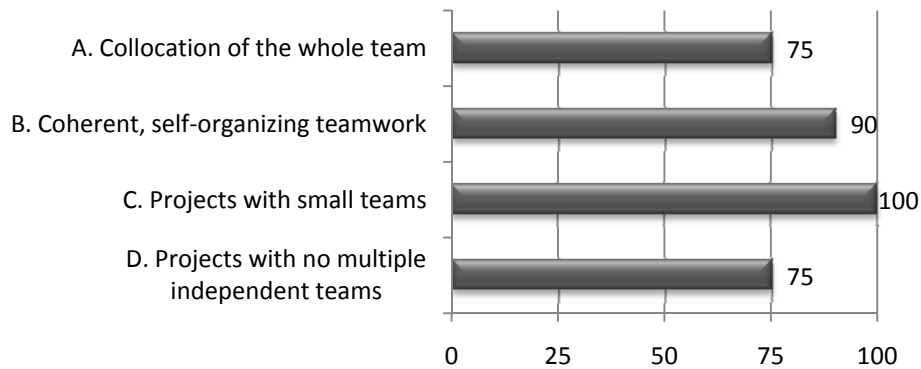


Figure 5:18 Audio Control Drivers of Team environment

As shown in Figure 5:19, the Team Leader's commitment and authority are total, and the relation with the end customer is good. Since the customer's development and management processes are extensive and including numerous other sub-projects within the own organization, it could hardly redesign them just to get aligned with the agile process practiced by Audio Control. With greater knowledge of Scrum it would, however, be easier for the customer to make adaptations. Though, with this kind of business relation, Audio Control cannot take full responsibility for training the customer.

Also in this business relation, and due to the nature of the work (90% support and maintenance), the target-cost approach is not applicable, and this driver has been left out of the analysis.

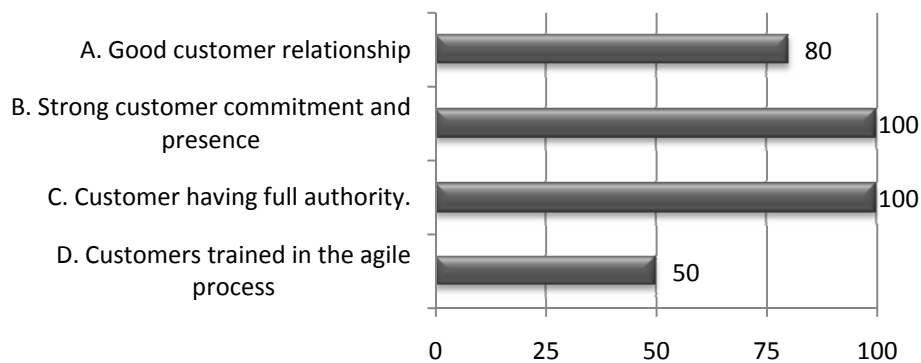


Figure 5:19 Audio Control Drivers of Customer involvement

ANALYSIS

The drivers that are least considered and have the lowest degree of implementation (a score under or equal with 50) are shown in Figure 5:20.

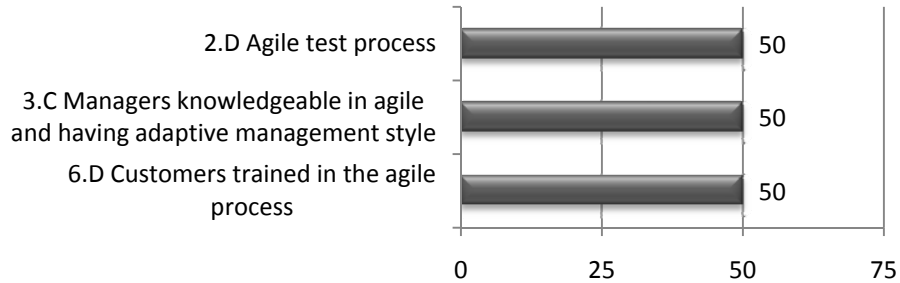


Figure 5:20 Least considered drivers in Audio Control

Other lower ranked drivers (with a score under 75) are presented in Figure 5:21.

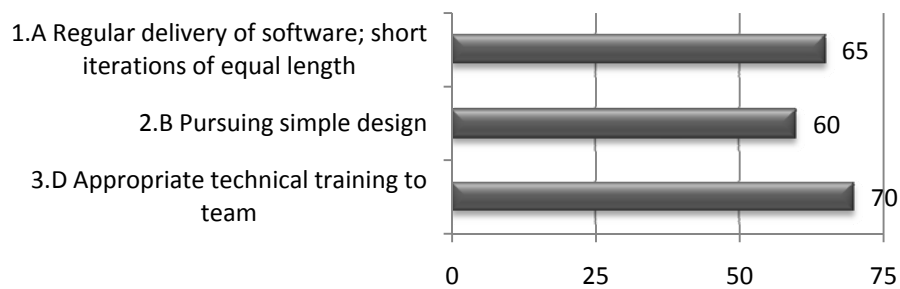


Figure 5:21 Other lower ranked drivers in Audio Control

The factor most significant, underlying driver 3.C, 6.D and 1.A, is the customer's inertia in adapting to the agile process, which may not be easy to improve. Other areas under Audio Control's control to improve are mainly within engineering techniques and to better align new development with the sprints.

ANALYSIS

5.3 CASE C – CASCADES AND KASTOR

5.3.1 Case C – Model reinforcement 3

The positions of Cascades' and Kastor's customers are clearly shown by the SCS described in chapter 5.3.2. It is also possible to collect and analyze relevant data in an unambiguous way without any need of adjusting the model. By being totally applicable also on this third case, the SCS model is further strengthened.

The CSF analysis for this case described below in chapter 5.3.3, is also reinforcing the CSF model by not needing any adjustments to make data collection and analysis explicit.

Driver *6.E Use of target-cost contracts to share risk* is not applicable for this case, and has not been applicable for any of the other cases in this study. This means that the driver is not reinforced, but does not imply that it should be removed from the CSF model. The benefits from target-cost contracts are well described in the literature, and for cases where these contracts are applicable it will be an interesting factor to analyze.

5.3.2 Customer classification

The customers of Cascades and Kastor could be described as *key customers*, *major customers* and *minor customers*. They all get the same product at the same time, but the key and major customers influence the development more by putting requirements on future releases. Though, none of the customer types are part of the agile process, and the transparency is minimal for each of them. This places all customers in the lowest part of the SCS matrix.

The customers may be aware that the products are developed with agile methodologies, but it is the road maps rather than the backlogs and sprints that are discussed. None of the customers attend any sprint demos or Scrum meetings, and the knowledge of other clients is limited (with exception to the public announced clients). This classifies all customers as Non-agile Indirect Participants.

The minor customers get what is released and seldom influences the development. They are therefore placed at the very left in the Non-agile-Indirect-Participants box, as shown in Figure 5:22.

ANALYSIS

The key customers do not get any customer-specific adaptations of the product, but influence the road map – and with that the development. This ability put them somewhat to the right in the box.

The major customers also put pressure on what to develop, but do not get as much attention for their requirements, and they are placed left of the key customers.

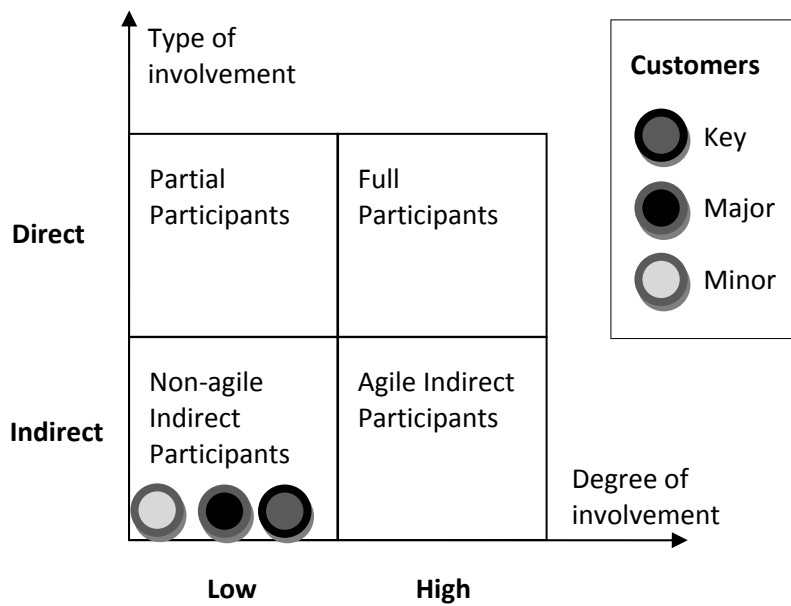


Figure 5:22 Position of Cascades' and Kastor's customers in the SCS

Future directions for involvement of the customers of Cascades and Kastor are hard to point out. The project is confident in the way the customers currently are approached, and experiences that the development process is well designed to meet the customer's wants and needs.

ANALYSIS

5.3.3 CSFs and Drivers

Since the customers are classified as Non-agile Indirect Participants, the Product Owner is considered to be the customer representative in the Scrum process.

The results of the driver investigation, for which the data is described in APPENDIX CASE STUDY DATA, are presented per CSF in Figure 5:23 and per driver in Figure 5:24.

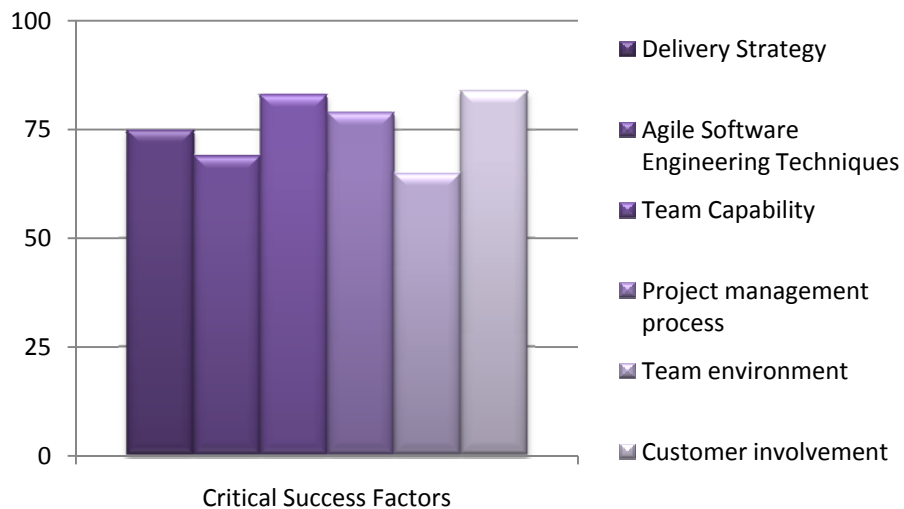


Figure 5:23 Implementation of the CSFs in Cascades and Kastor

Figure 5:23 shows that most drivers are implemented close to or better than *good*. The diagram points out that the second most important factor has the second lowest degree of implementation – but still at a good level. In the subsequent paragraphs the CSFs are analyzed individually.

ANALYSIS

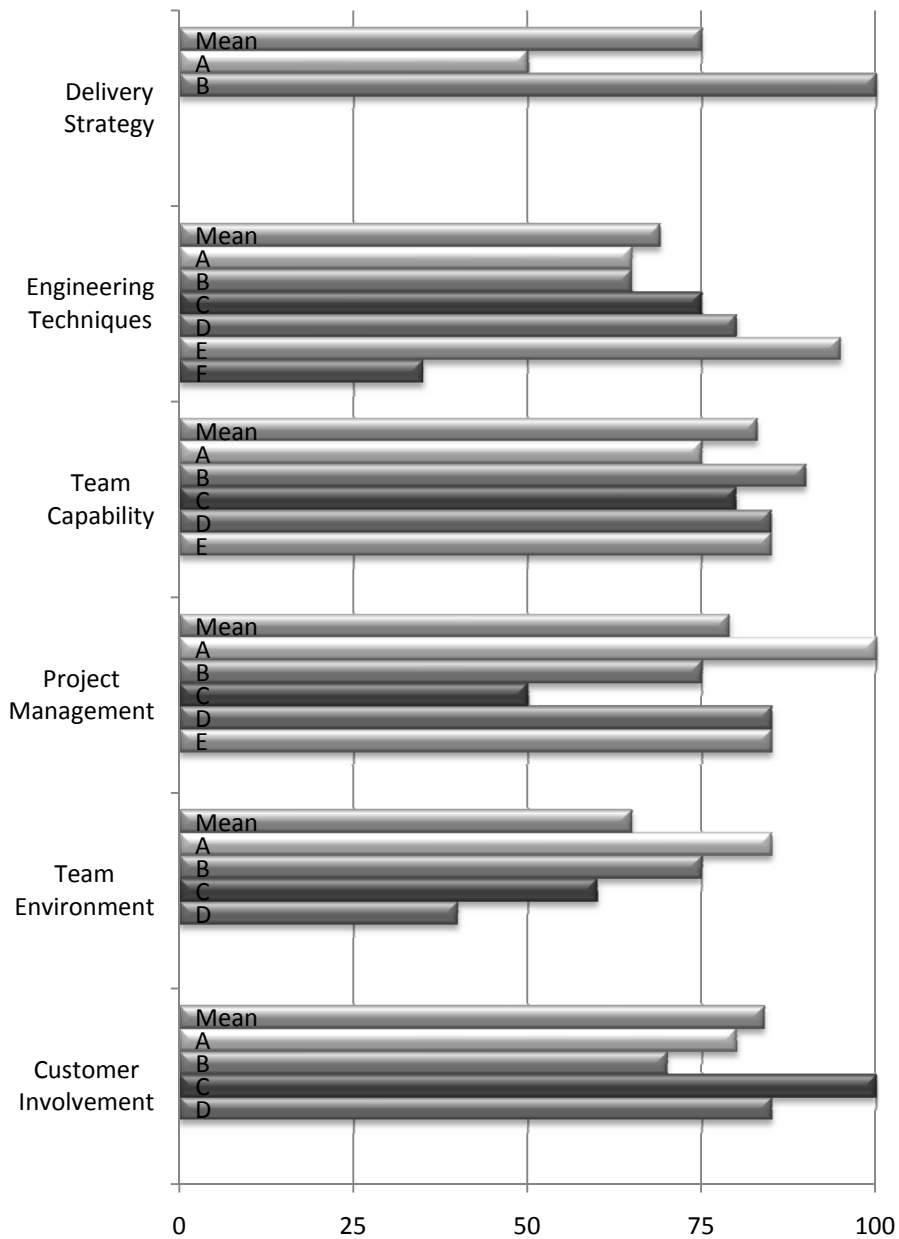


Figure 5:24 Implementation of each driver in Cascades and Kastor

The delivery strategy, visualized in Figure 5:25, is generally well implemented, and most important features are always developed and delivered first. Even if the sprint lengths are short, all tasks are not always finished within a sprint,

ANALYSIS

and criteria for when a task is complete are vague. Features sometimes take more than two weeks to develop, and not much attention is paid to demos.

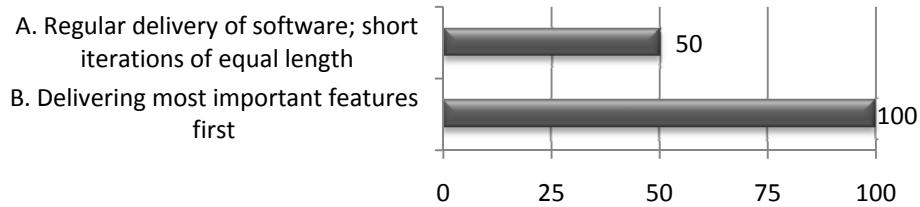


Figure 5:25 Cascades and Kastor Drivers of Delivery Strategy

Agile Software Engineering Techniques is the CSF with the second lowest scores. What stands out positively is the implementation of automatic tests and builds on a continuous integration server. In most areas at least a hygienic level is kept, but poor attention is paid to demos and the definition of when a task is done is vague. The design is often kept simple, and the few design documents that exist are published on a Wiki. Though, some of the more complex features, where functionality has been added gradually, could be better refactored. The drivers of Agile Software Engineering Techniques are shown in Figure 5:26.

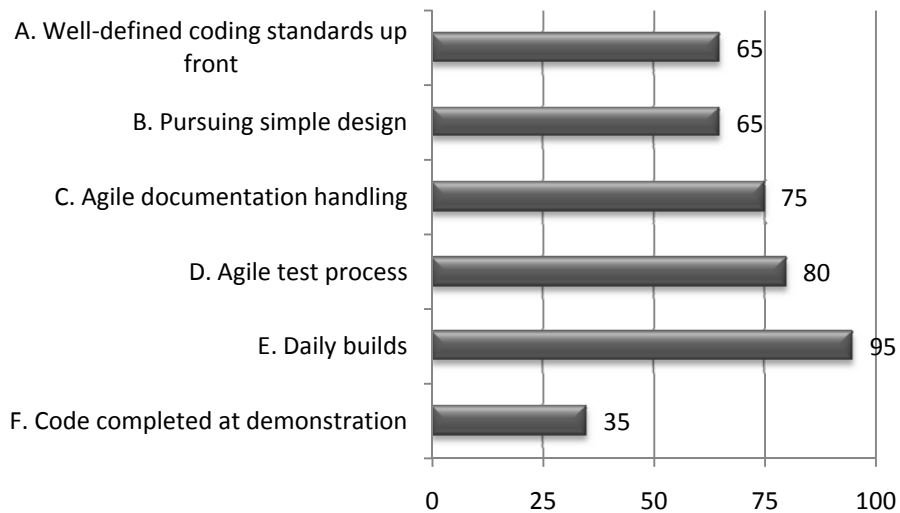


Figure 5:26 Cascades and Kastor Drivers of Agile Software Engineering Techniques

ANALYSIS

The measures of each driver for Team Capability are presented in Figure 5:27, and show that the project members are highly skilled, motivated, well trained and is committing to the project. Even if some people naturally are experts within certain areas, the team members try to share their knowledge with each other, and people with different roles are well complementing each other. It is not experienced that the individual expertise leads to bottle necks, but could be a risk. Most project members have a very positive attitude to the project and Scrum, while only a minority is indifferent. Also managers in direct contact with the project know Scrum well.

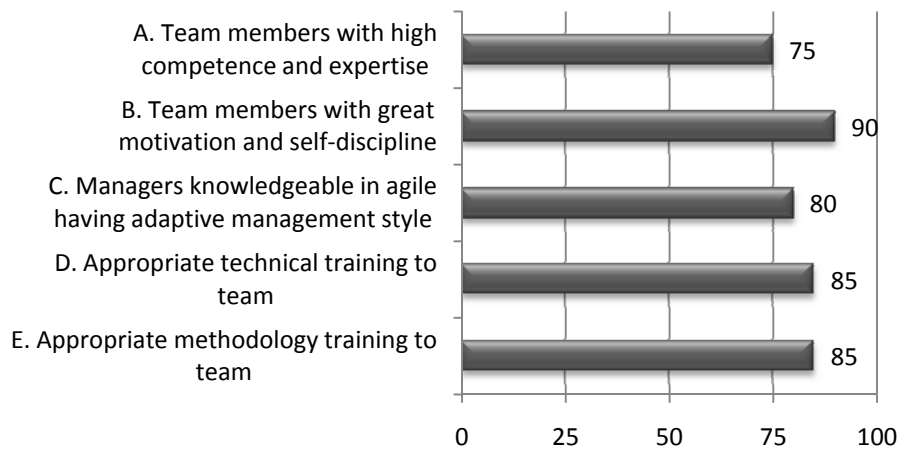


Figure 5:27 Cascades and Kastor Drivers of Team Capability

As shown in Figure 5:28, the requirement management process works very well, and the roadmaps for each product are translated to tasks in the PBLs by the Product Owner. At the sprint planning, the top priority tasks are put in the SBL, and distributed within the teams by the team members, meaning that the project-management process works well.

Also the communication focus is very strong, with daily Scrum meetings and co-location of the teams, and very little overtime is worked.

The project-management tool used in the project works rather good, but metrics are not very visible. As an example the burn-down charts were clearer when an ordinary spread-sheet was used. Velocity is not measured, but discussed to implement. The lack of visibility of the progress is not optimal for the project, but not critical since all features within a task seldom are developed.

ANALYSIS

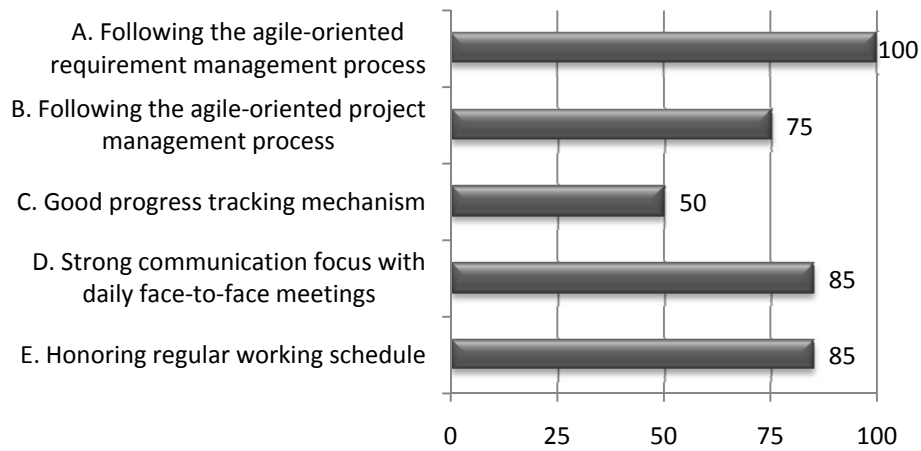


Figure 5:28 Cascades and Kastor Drivers of Project management process

Within the project each team is collocated, and both teams are located on the same floor together with the other related product teams. The Product Owner has his own office on the same floor close to the teams, and co-location is well implemented, as shown in Figure 5:29. There are many dependencies between the teams, concerning approximately 20% of the development, but problems are minimized by the close location and the temporary cross-product Feature Teams. Though, the dependencies often make it hard to finish tasks within a sprint, and location of the Feature Teams is not yet optimized.

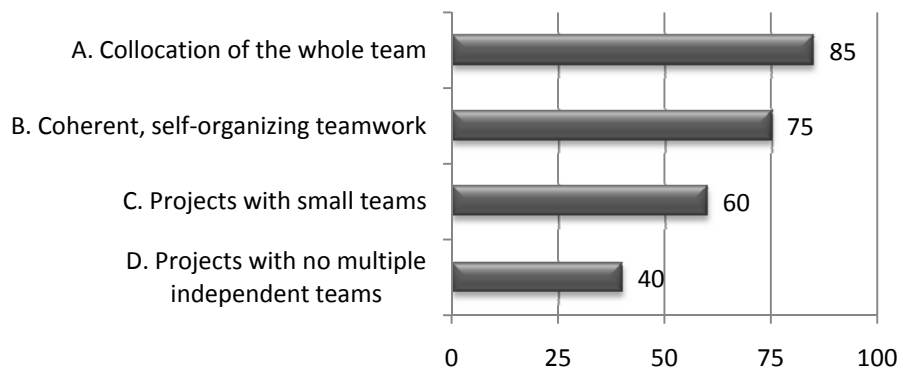


Figure 5:29 Cascades and Kastor Drivers of Team environment

ANALYSIS

The Feature Teams' ability to self-manage and focus on their tasks is experienced to be very good. The ordinary Scrum teams' ability is in general quite good, even if the self-managing approach not always may be the most efficient, at least not in the short run. This lack of efficiency could primarily be related to the slightly too large team; ten people in a team make it harder to keep a common focus on the tasks and the development. The project is aware of the problems with too large teams and many dependencies, and is planning for reorganizing with permanent Feature Teams.

As shown in Figure 5:30 the involvement of the customer – the Product Owner – is very high. He knows the agile methodology well and his prioritizations are followed. Though, he is located in his own office and attends only about one third of the meetings, keeping the presence factor down a bit.

Target-cost contracts are not applicable for Cascades and Kastor, since the customers are buying a standardized product.

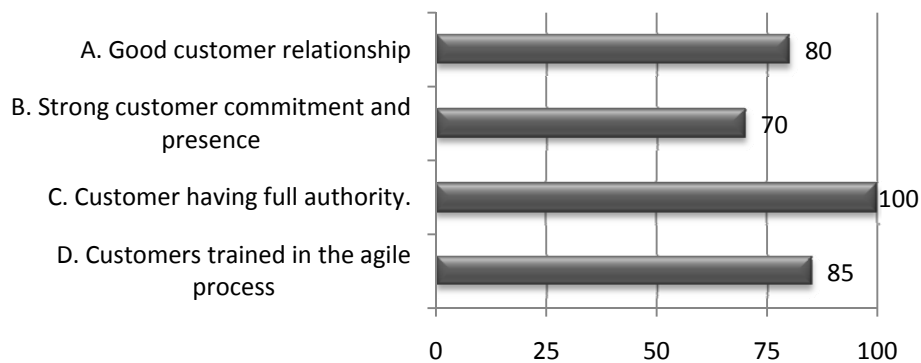


Figure 5:30 Cascades and Kastor Drivers of Customer involvement

ANALYSIS

The drivers that are least considered and have the lowest degree of implementation (a score under or equal to 50) are presented in Figure 5:31.

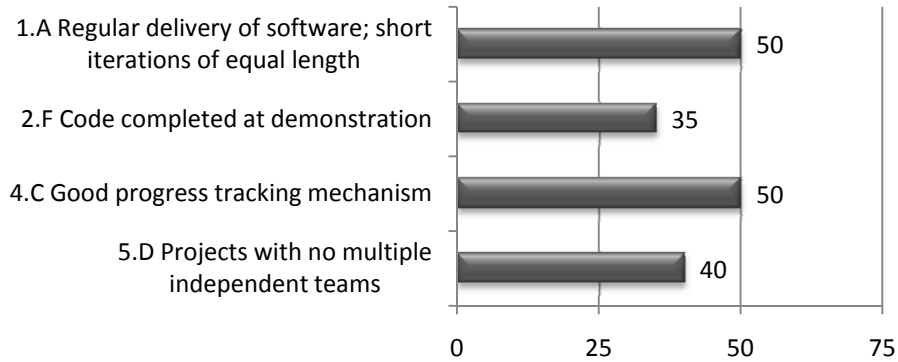


Figure 5:31 Least considered drivers in Cascades and Kastor

Other lower ranked drivers (with a score under 75) are described in Figure 5:32.

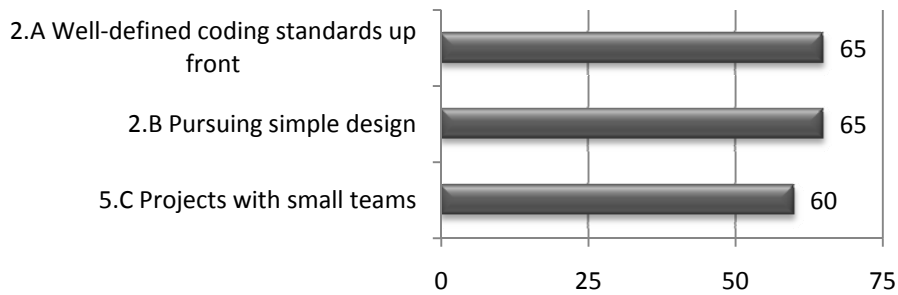


Figure 5:32 Other lower ranked drivers in Cascades and Kastor

By considering the drivers 1.A, 2.F, 4.C and 5.D, it is possible to further improve the development process, with the result that better products are developed. In this way both the internal working environment and the products will improve, and the perceived end-customer value increase.

ANALYSIS

5.4 REINFORCED MODELS

In this sub-chapter the reinforced – final – models are summarized.

5.4.1 Scrum Customer-classification System

In all three cases the positions of the customers are clearly shown by the SCS, without modifying the model. The classification of the customers by collecting and analyzing data can also be made unambiguously. When investigating the CSFs, the SCS classification is shown to give clear and necessary background information.

This means that the SCS as described in chapter 3.3 is reinforced and strengthened by the case studies.

5.4.2 Critical Success Factors

The original set of drivers, described in chapter 3.5, was adjusted in the first case study. By the following cases, where no adjustments were needed, the model is strengthened. The driver *6.E Use of target-cost contracts to share risk* has not been applicable in any of the cases, and is not reinforced. Though, as described in chapter 5.3.1, it should not be excluded from the model.

The reinforced CSF model, with the final set of drivers, is shown below in Table 5:2.

Table 5:2 Reinforced model of CSFs, Drivers and SCS customer types

CSFs and Drivers	Customer types			
Explanation of symbols: ● Driver is valid ○ Driver is valid with respect to in-house Product Owner – Driver is not valid	Full Participants	Partial Participants	Agile Indirect Participants	Non-agile Indirect Participants
1. Delivery Strategy				
A. Regular delivery of software; short iterations of equal length	●	●	●	○
B. Delivering most important features first	●	●	●	○

ANALYSIS

Table 5:2 (continued)

2. Agile Software Engineering Techniques				
A. Well-defined coding standards up front	●	●	●	●
B. Pursuing simple design	●	●	●	●
C. Agile documentation handling	●	●	●	●
D. Agile test process	●	●	●	●
E. Daily builds	●	●	●	●
F. Code completed at demonstration	●	●	●	●
3. Team Capability				
A. Team members with high competence and expertise	●	●	●	●
B. Team members with great motivation and self-discipline	●	●	●	●
C. Managers knowledgeable in agile and having adaptive management style	●	●	●	●
D. Appropriate technical training to team	●	●	●	●
E. Appropriate methodology training to team	●	●	●	●
4. Project management process				
A. Following the agile-oriented requirement management process	●	●	●	○
B. Following the agile-oriented project management process	●	●	●	●
C. Good progress tracking mechanism	●	●	●	●
D. Strong communication focus with daily face-to-face meetings	●	●	○	○
E. Honoring regular working schedule	●	●	●	●
5. Team environment				
A. Co-location of the whole team	●	●	●	●
B. Coherent, self-organizing teamwork	●	●	●	●
C. Projects with small teams	●	●	●	●
D. Projects with no multiple independent teams	●	●	●	●
6. Customer involvement				
A. Good customer relationship	●	●	●	○
B. Strong customer commitment and presence	●	○	○	○
C. Customer having full authority.	●	○	○	○
D. Customers trained in the agile process	●	●	○	○
E. Use of target-cost contracts to share risk	●	●	●	—

ANALYSIS

5.5 *BENCHMARK*

First in this section the main similarities and differences between the cases are summarized and their customer types are visualized. Then a comparison of their implementations of the CSFs and drivers is presented.

As the analysis models only are adjusted as a consequence of Case A, no case-analysis needs to be updated to be comparable with the others.

5.5.1 **Project backgrounds and customers**

General facts about the three cases are summarized in Table 5:3, highlighting main areas in the cross-case analysis in this sub-chapter, and referred to in the following paragraphs.

Mobical is the project with the most complex development process. The teams constantly have to deal with several customers' products all at once during the sprints, and requirements are added to the project from the Product Owner, the Customer Project Manager and the Program Manager. In Audio Control – the least complex project – only one customer exists, and in the Cascades and Kastor projects all customer requirements are handled by the Product Owner. He could then be seen as the one customer representative for whom one standardized product is developed.

In all cases each Scrum team has its own Scrum Master, but the role of the Product Owner differs. Both Mobical and Cascades and Kastor have an in-house Product Owner, but in Audio Control the role is divided between the Team Leader and the customer. Both Audio Control and Cascades and Kastor are organized with test as separate units, not included in the Scrum teams as in the Mobical case.

All cases experience structure as a benefit from Scrum. There are no other significant benefits common for all three of the cases, but common beliefs that visibility and efficiency also have improved due to Scrum exist. The two cases except Mobical also experience that communication has improved with Scrum.

ANALYSIS

Table 5:3 General comparisons of the cases

	Case A	Case B	Case C
<i>Type of development</i>	Product with adaptations for each customer	Outsourced development	Standardized product
<i>Customers</i>	Several customers of the same type	One customer	Key, major and minor customers
<i>Customer types</i>	Non-agile Indirect Participants	Partial Participants	Non-agile Indirect Participants
<i>Agile roles</i>	PO, CPM, PM and two SMs	TL	PO and two SMs
<i>Team organization</i>	Two Scrum teams where test is included	Two Scrum teams, separate test team	Two Scrum teams, separate test team
<i>Scrum implemented</i>	For 18 months	For 3 months	For 24 months
<i>Experienced benefits</i>	<ul style="list-style-type: none"> • Structure • No unnecessary planning • Prioritizations are visible 	<ul style="list-style-type: none"> • Communication • Visibility • Cooperation • Structure • More efficient work 	<ul style="list-style-type: none"> • Better structure • Communication • Improved customer-orientation
<i>Future potential benefits</i>	<ul style="list-style-type: none"> • In all areas • Increased commitment • Metrics 	<ul style="list-style-type: none"> • Efficiency • Metrics • Customer adapted to Scrum 	<ul style="list-style-type: none"> • Coordination • Efficiency • Quality • Metrics

ANALYSIS

The customer types for each case are shown in Figure 5:33, where it is visualized that Mobical and Cascades and Kastor have similar types of customers. Also shown is that both Mobical and Audio Control could gain future benefits by stronger customer involvement in the agile process.

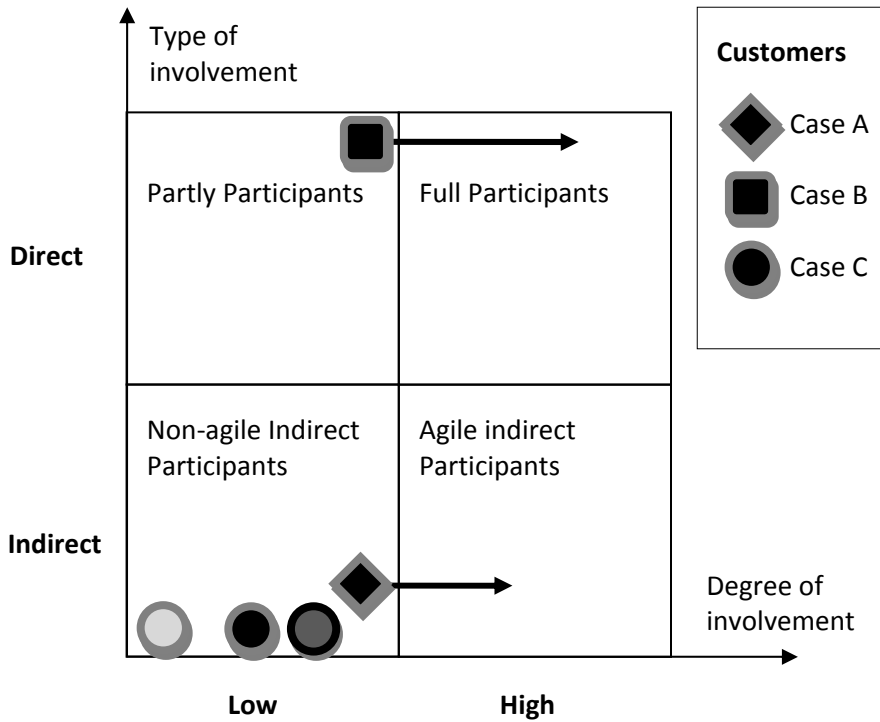


Figure 5:33 Position of customers of all cases in the SCS

5.5.2 Agile implementation

In Figure 5:34 implementation of the CSFs for all projects are shown. Even if the measures are not definitely comparable, it is indicated that Mobical – with the generally lowest degrees of implementation – should be able to improve its processes. It is also visible that the projects in general have lower implementation of the two or three most critical success factors than of the three less critical.

In the following paragraphs each CSF is compared between the cases.

ANALYSIS

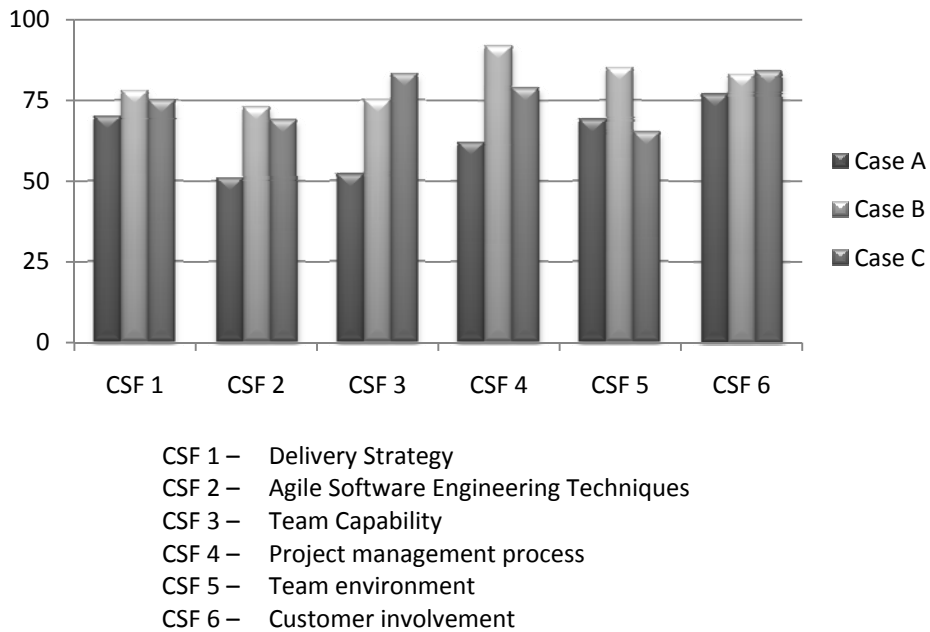


Figure 5:34 Implementation of the CSFs in all cases

The visualization of the delivery strategy in Figure 5:35 shows that all projects almost always implement the most important features first.

Mobical and Cascades and Kastor seldom finish all tasks within a sprint, pay low attention to sprint demos, and not often deliver increments of shippable functionality at the end of the sprints. Audio Control is better in following the sprint planning, and in finishing the tasks within the sprints.

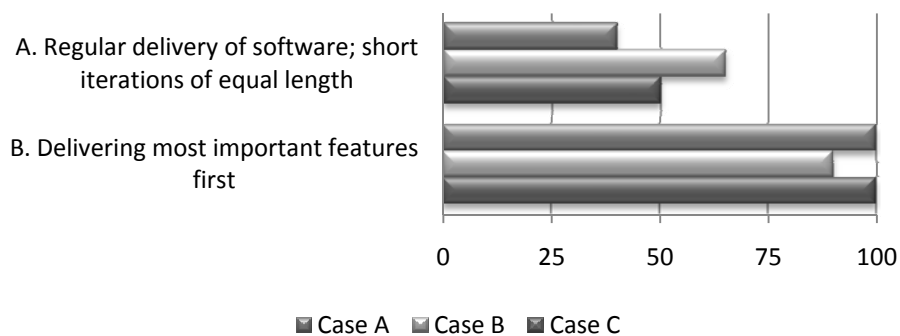


Figure 5:35 Drivers of Delivery Strategy

ANALYSIS

The drivers of Agile Software Engineering Techniques are shown in Figure 5:36. Coding standards are vaguely defined and followed in Mobical in opposite to the other cases, which are also better in keeping the code clean and simple. The Cascades and Kastor projects have a good implementation of automatic tests, which are not that well practiced in Mobical and Audio Control.

The most striking similarity between Mobical and Cascades and Kastor is that criteria for when a task is complete are vague or do not exist at all. This means that the code hardly could be completed at demonstrations. Audio Control, on the other hand, has well-defined criteria.

The implementation of driver *2.E Daily builds* is hard to compare, since the studied projects' systems are too different. Still it might be possible to improve – especially for Mobical.

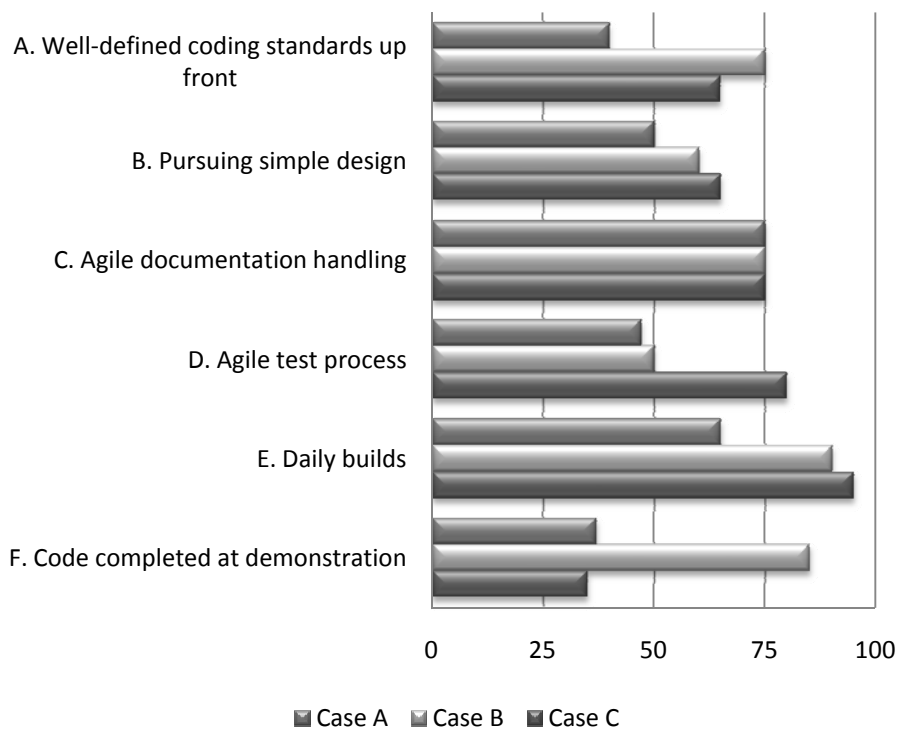


Figure 5:36 Drivers of Agile Software Engineering Techniques

ANALYSIS

As shown in Figure 5:37, Audio Control and Cascades and Kastor better deal with experts in the teams, and tasks – even within the expertise area – are divided between all developers. This may not always be the most efficient way, at least not in the short run, but gains common responsibility for the development, something that is lacking in Mobical.

Remarkable is that motivation is much lower in Mobical than in the other projects, which partly could be described as a result of different understandings of the practiced agile process. This driver is also related to methodology improvement and training, which has higher values in the other cases.

The knowledge of Scrum on a managerial level is hard to compare, since Mobical's value is kept down by a single person and managers for Audio Control are considered within the customer organization. Still it can be improved in both cases.

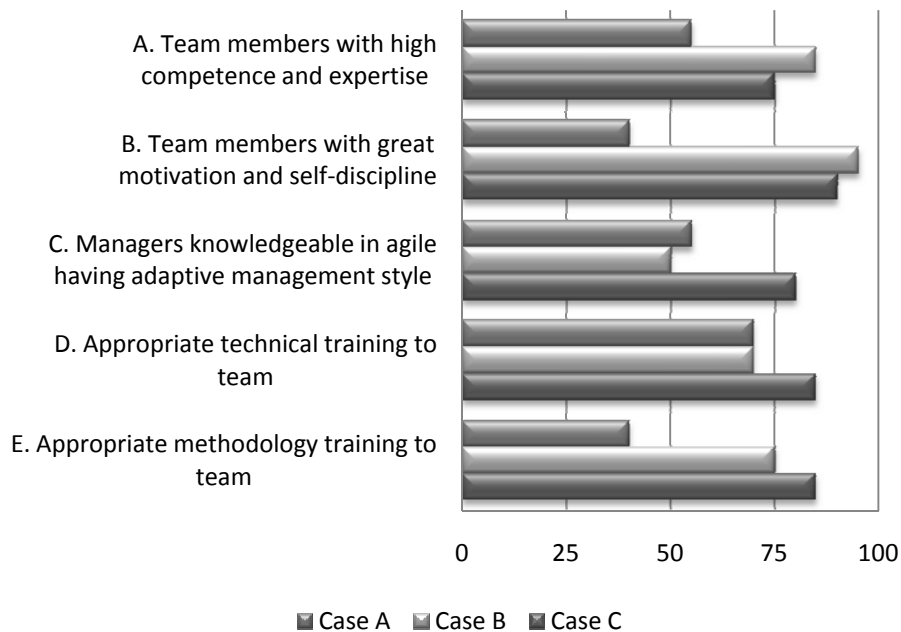


Figure 5:37 Drivers of Team Capability

ANALYSIS

Figure 5:38 shows that all projects follow the requirement management process well, and that they all have a strong communication focus. Overtime is in general related to releases in the Mobical case, and almost never worked in Audio Control and Cascades and Kastor.

The agile project management process is better implemented in Audio Control and Cascades and Kastor than in Mobical. In all cases the teams self-organize quite well at the sprint planning meetings, but Mobical are depending more on experts and do not tend to self-reorganize when tasks are added during the sprints.

None of the projects have found a really good project management tool, where reporting, metrics and follow up are easy and intelligible. The simplest tool, used by Audio Control, provides no metrics at all but neither requires much effort from the team members, and is good enough for their needs.

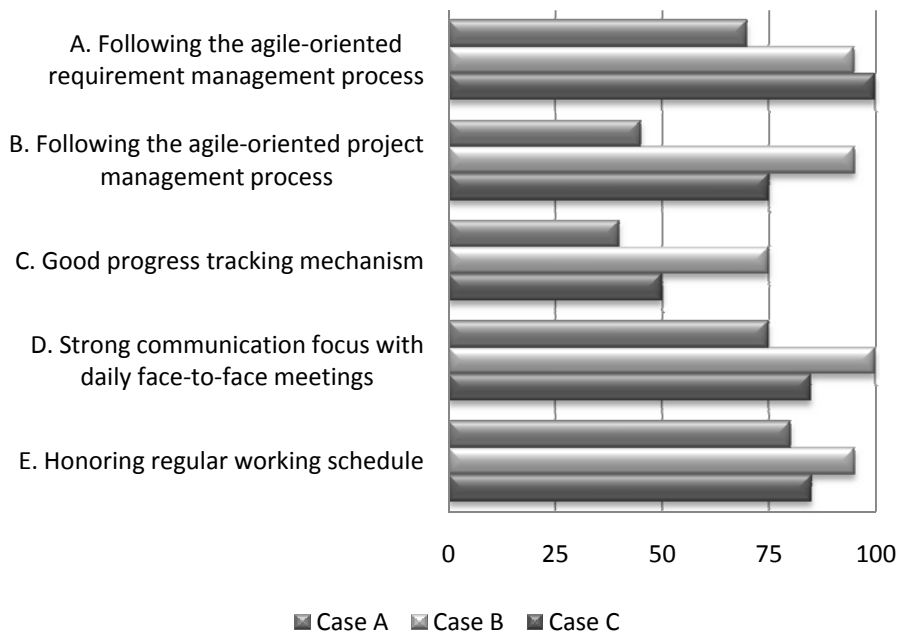


Figure 5:38 Drivers of Project management process

ANALYSIS

In Audio Control and Cascades and Kastor all project members are well collocated. This is the case in Mobical too, with exception to the Product Owner and particularly the Customer Project Manager. Their absence impacts the communication climate, and the driver stands out as shown in Figure 5:39.

Mobical and Audio Control are organized in small teams, which is not the case for both Cascades and Kastor teams. Though, common responsibility and self-organization are poor in Mobical.

In all projects the different Scrum teams focus on separate parts of the development, to minimize dependencies between the teams, but Cascades and Kastor stand out since the many dependencies affect the end-of-sprint deliveries.

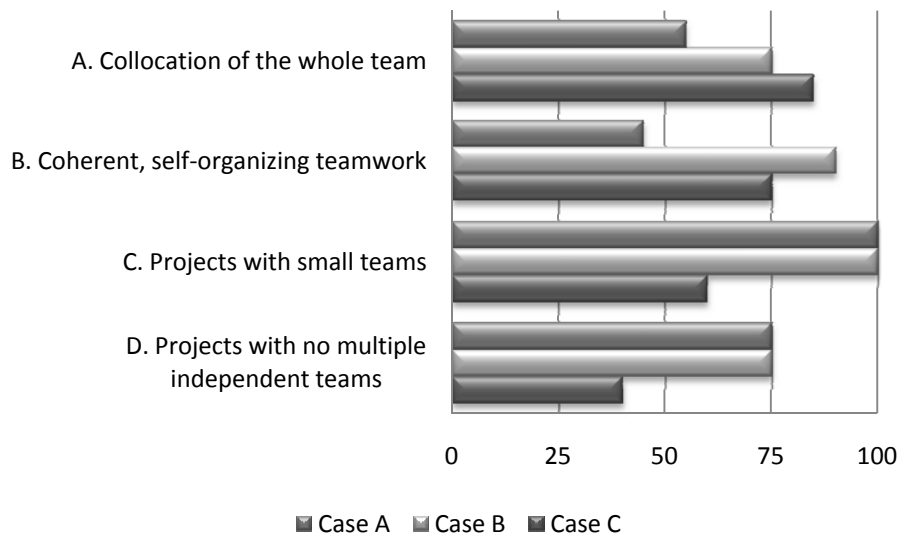


Figure 5:39 Drivers of Team environment

ANALYSIS

As shown in Figure 5:40, all projects have a good relation to their customer representative – the Product Owner – who also has full authority. Audio Control has the best commitment and presence, since the Team Leader is working within the team, while both the other cases have separate Product Owners.

If all of Mobical’s customer representatives had had a common view of the agile process, the measures of driver 6.D could have been as high as for Cascades and Kastor. This driver is not meaningful to compare with Audio Control, for which the end customer is referred.



Figure 5:40 Drivers of Customer involvement

In Figure 5:41 the implementation of the overall least considered drivers in Mobical (with a score under or equal to 50) are compared to the other cases. The drivers that stand out, i.e. are poorly implemented by Mobical alone, mainly consider the team members, their motivation and commitment, and their ability to manage themselves in the complex and hectic project.

Engineering techniques is another area which in general is little considered, and where the lack of coding standards stands out as a single practice compared to the other projects. Other areas, where the Cascades and Kastor projects also have low measures, regard vague definition of what *done* means and paying little attention to sprint deliveries.

ANALYSIS

Also the progress tracking and metrics in general are poorly implemented in Mobical as well as in Cascades and Kastor. This is also the case in Audio Control, but this project has not yet seen the need of more explicit metrics.

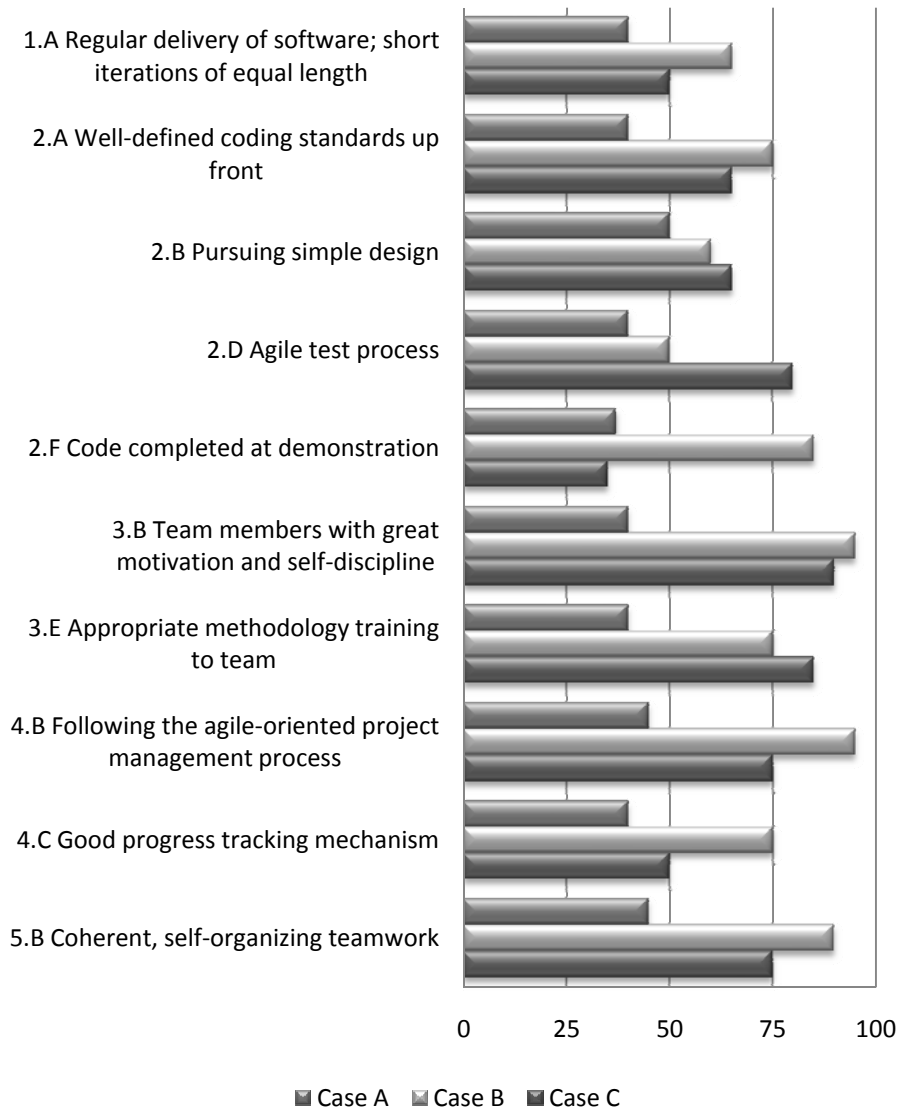


Figure 5:41 Least considered drivers in Mobical

ANALYSIS

5.5.3 **Benchmark summary**

The benchmark shows big differences in many areas and similarities in other. Some of the differences depend on individuals, some on the customer types and other on the team members' capabilities and commitment. For all cases two or three of the most significant CSFs are generally lower implemented than the other three, and overall Agile Software Engineering Techniques and Team Capability have the lowest scores.

The most complex project – Mobical – has on the whole the lowest degrees of implementation of the CSFs, while the least complex project – Audio Control – has the highest. The result and analysis of the interviews, together with general observations on the case sites, signify that the studied projects are marked by different levels of harmony; great in Audio Control, very good in Cascades and Kastor and at the moment not as good in Mobical.

6 DISCUSSION AND CONCLUSIONS

In this chapter the results from the analysis are discussed, initially related to the study propositions. Then the main question is discussed and conclusions are drawn. Finally the representativeness of the research and areas for future studies are discussed.

6.1 THE STUDY PROPOSITIONS

The study propositions, earlier described in chapter 2.3.3, are discussed one at a time in the following sections.

6.1.1 Drivers of customer value

The proposition discussed in this section is:

Which drivers of customer value can be identified in agile practices?

As described in chapter 3, by implementing the critical success factors a project is likely to be successful in matter of quality, scope, time and cost. Since these dimensions are equal to customer value, the attributes defining the CSFs are identified as drivers of customer value. The drivers describe methods, practices and environmental factors within a project, and measuring indicates how well they are implemented and highlights areas of improvement. By improving the practices underlying a driver, customer value will – direct or indirect – improve.

The initial set of drivers was expanded and explained in chapter 3, and have then been adjusted and reinforced when applied to the studied cases. The final set of drivers is presented in Table 5:2 *Reinforced model of CSFs, Drivers and SCS customer types* on page 86.

Since the success factors are ranked more or less critical, the drivers underlying the most significant CSF are the more important to consider. Though, it is not meaningful to specify to which level a more critical success factor should be implemented before considering improvements for less critical ones. Instead,

DISCUSSION AND CONCLUSIONS

for each individual project, practices to improve have to be chosen paying attention to specific conditions in the project. Still, the selection should be guided by the overall ranking of the CSFs.

6.1.2 Customer classification

This section discusses the following proposition:

How can customers be classified in matter of participation in an agile project?

In chapter 3.3 customers are identified to be part of the agile process either direct or indirect and their involvement to be graded as higher or lower. To visualize a customer's position, it can be allocated to any of the quadrants in the Scrum Customer-classification System. This classification of customers may be used when considering strategies for evolving business relations, and also provide necessary background information when applying the CSF model.

The SCS model is shown to be applicable for all the studied cases without adjustments, and is therefore reinforced. By also outlining relations between the different customer types, it indicates how customer relations could be evolved for the studied projects.

6.1.3 Benchmark

In this section the following proposition is discussed:

How well are the drivers and customer types considered by companies in Tactel's business context?

The SCS analysis of the different cases, put together in chapter 5.5.1, show that Mobical and Audio Control each have one single type of customer and that Cascades and Kastor have three. This is even though both Mobical and Cascades and Kastor have several single customers. Cascades and Kastor have the least agile relation to their customers, well suiting their needs when developing the same products for all customers without any customizations. Both Audio Control and Mobical identify possibilities to further involve their customers in the agile process, to gain benefits with the products fully developed or adapted for them.

Chapter 5.5.2 presents the benchmarking CSF analysis and shows how well agile practices are implemented in the different projects. Generally, it is shown that all cases pay less attention to the three significant critical factors than to

DISCUSSION AND CONCLUSIONS

the three less critical, and overall Mobical has the lowest degree of implementation. It is also shown that Audio Control on the whole has the highest scores, and since the level of complexity and workload are different in the three cases it is concluded that it is harder to implement the CSFs in a more complex project. Individual capabilities may also be a reason for the differences in implementation.

All projects have identified future potential benefits with Scrum and have an ambition to improve their processes, and one common area to improve is the use of metrics. Both Mobical and Cascades and Kastor also have unclear definitions of when a task is done and do not pay much attention to end-of-sprint deliveries. Drivers that are poorly implemented by Mobical alone mainly consider team members' motivation and commitment.

6.1.4 **Application on Mobical**

The proposition discussed in this section is:

How can Mobical benefit from the findings?

Mobical's customers are in chapter 5.1.2 identified as Non-agile Indirect Participants, but possibilities to involve them more in the agile process exist, making them Agile Indirect Participants. Though, since many different prioritized customers exist, the project would not benefit from increasing transparency and involving them even deeper – as Direct Participants.

Without referring to the other cases, a number of underlying factors critical for Mobical to improve are identified as follows. The three most evident and critical – based on the interview responses and the impact they have on several of the drivers – are presented first.

- the off-location of the Customer Project Manager and the poor communication between him and the team
- the absence of criteria for when a task is done
- team members' shortcomings in motivation
- the adding of too many tasks during the sprints
- the undefined engineering process
- team members' shortcomings in ability to self-organize and taking common responsibility
- the lack of metrics making it possible to follow up progress and improvements

DISCUSSION AND CONCLUSIONS

A very large effort should not be needed to create a common agreement of criteria for when a task is complete, and this factor is therefore strongly recommended to consider. The one of the studied cases that has defined these criteria is Audio Control; the development is finished and the code is reviewed, documented, built and sent to test. If the Customer Project Manager cannot be collocated with the team in Malmö, which easily should overcome most of the communication barriers, the project has to put effort in finding alternative solutions. The motivation of the team members is probably the hardest factor to improve, and effort has to be put in creating a better team-spirit.

Compared to the other cases Mobical has the overall lowest degrees of implementation of the CSFs, and in some areas Mobical certainly can learn from the others. On an organizational level, the other projects perform tests in separate teams outside the Scrum process. Testers and developers have different expertise, and having both types in the same Scrum team may restrain the common responsibility for the tasks. In all of the studied cases two (or more) related Scrum teams exist. All of them have tried to minimize the dependencies by focusing on different parts of the development; in Mobical and Audio Control as front-end and back-end and in Cascades and Kastor with one team for each product.

The implementation of the Scrum process in Mobical is not clear to all project members, and related to the other cases Mobical's project structure is more complex. The simplest implementation of Scrum is found in Audio Control, where communication is a good replacement for metrics and other practices. This agrees very much to the first statement in the agile manifesto; individuals and interactions over processes and tools. The use of metrics is considered to be evolved in both Mobical and in Cascades and Kastor, and by measuring velocity Mobical could get better in estimating time and planning the sprints. The many tasks added during sprints are also a problem for all cases. Mobical has approached this by estimating and reserve time for the unknown tasks by placeholders in the SBL. The other projects plan less of their time, and unforeseen tasks can be solved during the unplanned hours.

By how Scrum is done by the other projects and described by various sources, generally two ways are identified for Mobical to consider getting the Scrum process better understood and followed within the project; the simple way or the structured way. The simple way means to strip of all unnecessary practices, rely more on interpersonal interactions, decrease the workload for a few sprints, and somewhat start the implementation over. The structured way is to better define all exercises and put time in training the project members, bringing structure to the project. The former way is the more agile one, but

DISCUSSION AND CONCLUSIONS

some structure may be needed to bridge shortcomings in motivation and common responsibility.

Irrespective of which way to follow, Mobical could reflect on further areas that differ between the studied cases; the automatic tests and builds very good implemented and visualized in Cascades and Kastor, and as in the same case engage an external Scrum consultant to inspire the project members. Mobical could also consider if it is possible to better standardize the product, and perhaps separate the standard-product development from the work with the many customizations in the agile process. The Cascades and Kastor teams' standardized products, with releases every three months, are developed in a less stressful way, and with more high level structure in the development process.

6.1.5 **Success of the proposition study**

Chapter 2.3.3 outlines criteria for the propositions to meet to be judged successful, which are referred to in the following paragraphs.

Since drivers of customer value are identified, and the CSFs are ranked by importance, the study of the first proposition could be said to be successful. It is, however, not possible to rank each individual driver in a strict hierarchy, but the groups of drivers representing the CSFs are.

By the SCS customers' participation in the agile process can be classified. The model also shows relations between the different types, and the success criteria are met.

This research is presenting how different customer types and implementation of the CSFs are considered by the studied cases – not by all competitors in the business. Though, it shows how Scrum is practiced in the studied projects, and that all of them are well aware of how to treat their different customer types. Accordingly, the success criteria for the third proposition are met.

Mobical's customer type is identified by the SCS, and drivers and several underlying factors for the project to improve are identified. Mobical's agile process is also compared to the other cases, and specific practices and general areas are put side by side to highlight differences and similarities. Hence, the success criteria for the fourth proposition are also met.

Since the success criteria are met for all four propositions, they are considered to be well answered. This means that they serve as a good foundation in the

DISCUSSION AND CONCLUSIONS

discussion of the main research question below, and clearly connect the results to Tactel and Mobical.

6.2 *THE MAIN QUESTION*

The main research question discussed in this section is:

How and where can Scrum be lifted from an internal management tool to get the customers more involved, and where is customer value created?

Where – for which types of customers and projects – Scrum can be evolved to engage the customers more is related to the Scrum Customer-classification System and the Benefits/cost ratio model for customer value. The SCS model, set up in the theory section and proved applicable by the cases, outlines four general types of customers; Full Participants, Partial Participants, Agile Indirect Participants, and Non-agile Indirect Participants. Except for the very most Full Participants, all customers could be deeper involved in the agile process; e.g. by learning about it, by taking part in continuous re-prioritizations, by attending meetings, and by undertaking the role of the Product Owner. Though, all of these steps are not always gaining business value. When a project develops a product for more than one customer and the customers not are transparent for each other, they might be contributing with requirements to consider in a road map or even prioritizing tasks in an own product backlog, but they could not have the role of the Product Owner. This is, accordingly, depending on the type of the project, and as stated in the SCS there are major differences between the direct and indirect participant customers.

Even if a customer could be deeper involved due to project characteristics, it also must expect to benefit from committing to the process. If the customer experiences that the total sacrifices are larger than the total benefits, customer value will not increase but decrease. Therefore, the types of customers that could be more involved in the agile process are the ones expecting benefits with a deeper engagement, and deeper involvement is possible for all kinds of projects – but to different extent.

How – which factors that should be considered – also depends on the customer's position in the SCS. As described above, different customer types can be involved in various ways, which also means that the possibilities to progress are dissimilar. When increasing a customer's degree of involvement it

DISCUSSION AND CONCLUSIONS

may be transitioned to another customer-type group in the SCS, as shown in Figure 3:5 *SCS hierarchy*. Since a customer is classified by the characteristics of its present involvement, the characteristics of the “next” level in the hierarchy are to be considered getting the customers deeper engaged. If the characteristics are applicable, due to the project type and the customer’s expectations of sacrifices or benefits, customers can be more involved.

Customer value is created where the customer experiences it. Chow and Cao’s study (14) outlines a set of critical factors to implement, considering the right product for the right price at the right time. For a project to ensure success in matter of customer value, the factors should be considered. These critical success factors are in this research evolved in the CSF model and proven applicable on the cases. This means that customer value – direct or indirect – is created by the practices underlying the CSFs – the drivers of customer value.

6.3 *FINAL CONCLUSIONS*

How and where Scrum can get the customers more involved in the agile process, and where customer value is created is discussed and answered above. The conclusions from the discussion are that customer value is created by the practices underlying the CSFs, the characteristics of the higher level in the SCS hierarchy describe how to get a customer deeper engaged, and that customers should be more involved only when they are expecting benefits with a deeper engagement. Though, the extent of the involvement depends on characteristics of each individual project.

Above is also discussed how Mobical can benefit from this study by considering a better implementation of a number of factors. Some of these factors concern the off-location of the Customer Project Manager, the absence of criteria for when a task is done and the need of getting the team members more motivated. Taking the outlined factors into account will help Mobical creating a better product, with the result that customer and business value are increased.

6.3.1 **Generalizations and limitations**

In this study only a few factors have negative impact on the quality of the research, leading to limitations in matter of reliability, validity and representativeness. The identified deficiencies concern the narrow selection of cases and presumed bias from some of the interviewees.

DISCUSSION AND CONCLUSIONS

The part of the study specifically focusing on Mobical is not to be generalized, and the results are applicable on Mobical only. Though, the results of the benchmark could be used in benchmarks of other projects meeting the same criteria as described in chapter 4.2. The benchmark itself compares Mobical to two individual cases selected to get a wider picture of implementations of Scrum, but it could not be said to represent all competitors in the business since the additional cases are selected from a narrow population. This means that the validity of the study is restricted in a wider context, limiting the representativeness of the research.

Even though bias is tried to be detected, some of the interviewees may have provided misrepresenting data, with the result that the CSF analysis generally shows too high or too low results. This makes a direct comparison of critical success factors and drivers between projects irrelevant.

The SCS model is verified to be applicable for rather small software development Scrum projects, with about one to five Scrum teams. Still no contradictions for the SCS to be used in the analysis of larger projects are found by this research, and the model could also be applicable for projects run by other agile methods, but might then need slight adaptations. The relevance of the SCS for agile projects not developing software is not studied; neither are any problems identified for applying the model.

An area not discussed in this research is how to evaluate if a customer would experience overall benefits from taking a greater part in the development process. Neither is how to promote the agile advantages, making the benefits with a deeper engagement clear for the customer. This should be kept in mind when considering the SCS and approaching a customer for deeper involvement.

The CSF model is shown to be applicable for the studied cases, and by its theoretical ground it could also be applicable for all agile software development projects independently of size and methodology. Since many of the CSFs and drivers consider specific software engineering techniques, the model is not valid for non-software projects.

The results of the study, in matter of the main research question, therefore are representative for small software development Scrum projects. The research also indicates that the results are able to be generalized for larger projects, and with adaptations also to projects run by other agile methods.

DISCUSSION AND CONCLUSIONS

6.3.2 **Areas for further research**

Further research could be done to study the validity of the SCS and CSF models for larger projects and for projects managed with other agile methods than Scrum. The reliability of the results could as well be increased by a quantitative approach, and they could be made representative in a wider context if a larger business segment is studied. These kinds of studies could then lead to a theorization of the models.

It also would be of interest to generally study projects working with agile methods outside the software development business, and specifically study where customer value is created in these projects.

How to evaluate if a customer would experience overall benefits from taking greater part in the development process, and how to promote the advantages, might as well be subjects for future studies.

Tactel and Mobical would benefit from following up the results of the analysis to measure how the project progresses. The first follow-up could be done when the most critical factors are corrected, and then analysis of the CSFs could be done on a regular basis.

1. *Empirical studies of agile software development: A systematic review.* **Dybå, T. & Dingsøy, T.** 9-10, 2008, Information and Software Technology, Vol. 50, pp. 833–859. Issn 09505849.
2. **Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.** *Manifesto for Agile Software Development.* [Online] 2001. [Cited: October 8, 2008.] <http://agilemanifesto.org>.
3. *Managing Agile Projects.* **Augustine, S.** London : Prentice Hall PTR, 2005. ISBN 0131240714.
4. **Tactel AB.** Main. *Tactel AB.* [Online] 2008. [Cited: September 24, 2008.] <http://www.tactel.se>.
5. **Höst, M., Regnell, B. & Runeson, P.** *Att genomföra examensarbete.* Lund : Studentlitteratur, 2006. ISBN 9144005210.
6. **Yin, R. K.** *Case study research.* 3rd ed. s.l. : Sage Publications Ltd., 2003. ISBN 9780761925538.
7. **Sellstedt, B.** *Metodologi för företagsekonomer.* September 2002. s.l. : Handelshögskolan i Stockholm, 2002. Vol. 2002:7.
8. *Checklists for Software Engineering Case Study Research.* **Höst, M. & Runeson, P.** s.l. : IEEE, 2007. First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007). pp. 479-481. Issn 19386451.
9. *Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research.* **Erickson, J., Lyytinen, K. & Siau, K.** 4, 2005, Journal of Database Management, Vol. 16, pp. 88-100. Issn 10638016.
10. **Schwaber, K.** *Agile Project Management with Scrum.* Redmond : Microsoft Press, 2004. ISBN 9780735619937.
11. **Schwaber, K.** *The Enterprise and Scrum.* Redmond : Microsoft Press, 2007. ISBN 9780735619937.
12. *A case study on the impact of customer communication on defects in agile software development.* **Korkala, M., Abrahamsson, P. & Kyllönen, P.** Minneapolis, MN : s.n., 2006. Agile Conference. 11 pp.
13. *Customer value: a review of recent literature and an integrative configuration.* **Khalifa, A. S.** 5-6, 2004, Management Decision, Vol. 42, pp. 645-666. Issn 00251747.

REFERENCES

14. *A survey study of critical success factors in agile software projects.* **Chow, T. & Cao, D.B.** 6, 2008, *The Journal of Systems and Software*, Vol. 81, pp. 961-971.
15. **Berteig, M.** Agile Benefits: Satisfied Stakeholders. *Agile Advice*. [Online] September 17, 2007. [Cited: October 14, 2008.] <http://www.agileadvice.com/2007/09/17/agilemanagement/agile-benefits-satisfied-stakeholders/>.
16. **Elssamadisy, A.** *Patterns of Agile Practice Adoption – The Technical Cluster*. s.l. : C4Media, 2007. ISBN 9781430314882.
17. *Factors that Significantly Impact the Implementation of an Agile Software Development Methodology.* **Livermore, J. A.** 4, 2008, *JOURNAL OF SOFTWARE*, Vol. 3, pp. 31-36.
18. *The XP customer role in practice: three studies.* **Martin, A., Biddle, R. & Noble, J.** 2004. Agile Development Conference. pp. 42-54.
19. *A case study on the impact of scrum on overtime and customer satisfaction.* **Mann, C. & Maurer, F.** 2005. Agile Development Conference. pp. 70-79.
20. *Selling agile: Target-cost contracts.* **Eckfeldt, B., Madden, R. & Horowitz, J.** 2005. Agile Development Conference. pp. 160-166.
21. **TAT, The Astonishing Tribe, AB.** *The Tribe. TAT.* [Online] 2008. [Cited: November 14, 2008.] <http://www.tat.se/>.

APPENDIX CASE STUDY DATA

This appendix describes the investigation protocols used for data collection in the cases. In the second sub-chapter it also presents the data collected and used in the analysis.

INVESTIGATION PROTOCOLS

The protocol described in Table A:1 is used for collecting data for the areas *Background* and *Customer characteristics*. Table A:2 describes the protocol used for collecting data for the *Drivers of customer value*. Presented are the final protocols, describing data collection according to the final models of SCS and CSFs.

Beside these protocols, notes have been taken on observations and additional information provided by the interviews, which for each case have been taken into account when compiling the records in the database.

APPENDIX CASE STUDY DATA

Table A:1 Investigation protocol – Part 1

BACKGROUND	
1	General about the company
1.1	Are there any limitations in publishing the collected data?
1.2	Name of the company?
1.3	Founded?
1.4	What types of products does the company develop?
1.5	Who are the major customers for the company?
1.6	How many employees?
1.7	Multinational?
1.8	Does the company have experience of agile methodologies in several projects?
2	
2	General about the project
2.1	Name of project
2.2	What types of products does the project develop?
2.3	How could the project be described in the company's organization?
2.4	Who are the major customers for the project?
2.5	Is the project using any project management tool?
3	
3	Agile project specifics
3.1	Which agile method or methods are in use?
3.2	For how long has the project been managed with agile methodologies?
3.3	To what extent is the project following the agile practices "by the book"?
4	
4	Group organization
4.1	Total number of people in the project?
4.2	Agile Roles in the project? <ul style="list-style-type: none"> • Product Owner • Scrum Master • Team
4.3	Support Roles in the project? <ul style="list-style-type: none"> • Marketing organization • Customer project manager • Technical support • Other
4.4	Who in the project are responsible for customer relations?
CUSTOMER CHARACTERISTICS	
5	
5	General customer characteristics
5.1	Does the project have one or multiple customers? <ul style="list-style-type: none"> • How many? • Are there differences between them in size and importance?
5.2	Does the project have a separate backlog for each customer? <ul style="list-style-type: none"> • If not, how does the project handle the requirements?

APPENDIX CASE STUDY DATA

6	Customer involvement
	This set of questions relates to a single customer. If several customers are analyzed, they should be repeated for each. With <i>customer</i> means in this section the end customer – not an in-house Product Owner.
6.1	Is the customer holding the role of the Product Owner?
6.2	Is the customer owning and prioritizing the Product Backlog?
6.3	Is the customer acting as a project member?
6.4	Is the customer collocated with the team?
6.5	Are the project and all its other customers transparent to this customer?
6.6	How well does the customer know the agile methodology? (0-100%)
6.7	Is the customer allowed to attend any meeting?
6.8	Is the customer attending the sprint reviews?
	EXPERIENCED AND EXPECTED BENEFITS
7	Benefits
7.1	Which main benefits is the project experiencing by the agile methods? <ul style="list-style-type: none"> • Improved customer collaboration • Reduced time to market • Increased value to market • Increased flexibility • Improved work process for handling defects • Higher quality (less support errands) • Improved estimation • Decreased overtime • Other
7.2	In which areas does the project identify future potential benefits? <ul style="list-style-type: none"> • Improved customer collaboration • Reduced time to market • Increased value to market • Increased flexibility • Improved work process for handling defects • Higher quality (less support errands) • Improved estimation • Decreased overtime • Other

APPENDIX CASE STUDY DATA

Table A:2 Investigation protocol – Part 2, CSFs and Drivers

The following steps are followed when investigating each driver:	
<ol style="list-style-type: none"> 1. It is analyzed if the driver is considered in the project, and how the project is working with it. 2. It is analyzed if it is having effect. 3. The degree of implementation is estimated, and a preliminary measure is set. 4. If the degree of implementation is low, it is analyzed if the project is aware of the driver, i.e. if it has been neglected by purpose. And if so, why it has been neglected. 5. A value, 0-100%, is estimated for each driver, based on the interviewees' opinions and the studier's observations and conclusions. 	
1. Delivery Strategy	
A. Regular delivery of software; short iterations of equal length	
B. Delivering most important features first	
2. Agile Software Engineering Techniques	
A. Well-defined coding standards up front	
B. Pursuing simple design	
C. Agile documentation handling	
D. Agile test process	
E. Daily builds	
F. Code completed at demonstration	
3. Team Capability	
A. Team members with high competence and expertise	
B. Team members with great motivation and self-discipline	
C. Managers knowledgeable in agile and having adaptive management style	
D. Appropriate technical training to team	
E. Appropriate methodology training to team	
4. Project management process	
A. Following the agile-oriented requirement management process	
B. Following the agile-oriented project management process	
C. Good progress tracking mechanism	
D. Strong communication focus with daily face-to-face meetings	
E. Honoring regular working schedule	

APPENDIX CASE STUDY DATA

5. Team environment	
A. Co-location of the whole team	
B. Coherent, self-organizing teamwork	
C. Projects with small teams	
D. Projects with no multiple independent teams	
6. Customer involvement	
A. Good customer relationship	
B. Strong customer commitment and presence	
C. Customer having full authority.	
D. Customers trained in the agile process	
E. Use of target-cost contracts to share risk	

APPENDIX CASE STUDY DATA

APPENDIX CASE STUDY DATA

COLLECTED DATA

This section serves as a database of the case studies, and presents condensed data from each case separately.

Case A – Mobical

Investigation date: 2008-11-03
 Interviewees: Scrum master/test leader
 Scrum master/developer
 Former Scrum Master/Program Manager
 Product Owner.

Table A:3 Mobical Investigation protocol – Part 1

BACKGROUND	
1	General about the company
1.1	Are there any limitations in publishing the collected data?
	Names of customers and figures are not to be published, but are not of importance for the study.
1.2	Name of the company?
	Tactel AB
1.3	Founded?
	Mid 1990s
1.4	What types of products does the company develop?
	Tactel is a developer of mobile applications, providing solutions and consulting services to many of the world’s major operators and handset vendors
1.5	Who are the major customers for the company?
	(see above)
1.6	How many employees?
	About 350
1.7	Multinational?
	Offices in Sweden, USA, Ukraine. Customers globally.
1.8	Does the company have experience of agile methodologies in several projects?
	In some, Mobical have worked longest according to agile methods.

APPENDIX CASE STUDY DATA

2	General about the project
2.1	Name of project
	Mobical
2.2	What types of products does the project develop?
	Services for synchronizing and backing up data on mobile phones. Most often is the service sold in a first phase as a standardized product, with customizations for the customer. In later phases more customer-unique adaptations and features are developed.
2.3	How could the project be described in the company's organization?
	The company is divided in subunits, each managing several development projects. This study is mainly focusing on the Mobical development team located at the Malmo office.
2.4	Who are the major customers for the project?
	Main customers are network operators and service providers worldwide, offering the service to their subscribers.
2,5	Is the project using any project management tool?
	Hansoft for administrating backlogs, burn-down charts, allocate people etc.
3	Agile project specifics
3.1	Which agile method or methods are in use?
	Scrum
3.2	For how long has the project been managed with agile methodologies?
	About 18 months.

APPENDIX CASE STUDY DATA

3.3	<p>To what extent is the project following the agile practices “by the book”?</p> <p>50/50</p> <p>Follows The major parts; PBL, SBL, 2-week sprints, the planning process, the meeting structure (Daily Scrums, Sprint Retrospective, Demonstration etc.), team sizes of five people in each team.</p> <p>Don’t follow New tasks are added during a sprint (e.g. support errands and sale support), since they are too urgent to be postponed for the next sprint. This is partly solved by the use of placeholders in the sprint planning. All tasks are not first specified in the PBL and then lifted into the sprint, but CPM and others contact individual team members for sale support, hot fixes etc, without any tasks written.</p> <p>Deliveries are made during the sprints, and not only after each sprint, and are planned for by tasks in the backlogs.</p> <p>Stories – visualization of what to develop – are not put together. They could be used to define high-level requirements in the PBL, and then broken down in tasks.</p> <p>The use of metrics is poor, for example velocity.</p> <p>Additional roles exist; CPM and PM. The CPM acts as support for the PO and the PM as a line manager. Their roles are somewhat unclear to the team.</p> <p>General reflections Many tasks are not finished in the sprints. This may depend on the complex system and underestimated lead times for task-switching.</p> <p>Some project members are not as committed to Scrum as others.</p> <p>The end customer is not involved in the process, but the PO acts as a customer proxy.</p>
-----	--

APPENDIX CASE STUDY DATA

4	Group organization
4.1	Total number of people in the project?
	<p>Mobical is organized in two Scrum teams – front-end and back-end – with focus on different parts of the system, and with about five engineers (developers and testers) in each. In both teams, one of the team members also acts as Scrum Master.</p> <p>The PO owns the roadmap for the product, and owns and prioritizes the PBL. He is also responsible for selling the product, and have to his support a CPM and a seller. The PO and the seller have generally the initial contacts with the customers, defining the high-level scope. Then the CPM works through the details with the customer, specifying detailed requirements and deadlines, and is their main contact during the project.</p> <p>One of the team members also works as PM, which is similar to a line management role. He owns the development, and is responsible for team competence, resource allocation etc.</p> <p>The PO is main responsible for adding tasks and prioritizing the PBL, but the CPM and the PM also edits it.</p> <p>To improve the development process, evaluating groups exist. One for each of test, development and sale, and one comprehensive group with two people from each of the sub-groups. They meet about every week. I refer to these as Process Improvement Meetings (PIMs).</p> <p>A group of developers and testers located in Ukraine supports the main project by developing and testing independent parts of the products. They are not included in the agile process, and not further considered in this case study.</p>
4.2	Agile Roles in the project?
	(see above)
4.3	Support Roles in the project?
	(see above)
4.4	Who in the project are responsible for customer relations?
	The PO and CPM and seller are responsible for customer relations.

APPENDIX CASE STUDY DATA

CUSTOMER CHARACTERISTICS	
5	General customer characteristics
5.1	<p>Does the project have one or multiple customers?</p> <ul style="list-style-type: none"> • How many? • Are there differences between them in size and importance?
	<p>Mobical has about fifteen recent customers. When a system is delivered to a customer a support-phase begins, and the work could be referred to as development or support. The sales organization is also selling new features and updates to the existing customers.</p> <p>Some customers are larger and more important than others. Support errands from major customers are prioritized during sprints, and not always put in the PBL.</p> <p>Mobical is about to finish a partner deal with a major service provider (hereafter called MSP), marketing and further developing Mobical's services. The MSP will then become one of the more important customers.</p>
5.2	<p>Does the project have a separate backlog for each customer?</p> <ul style="list-style-type: none"> • If not, how does the project handle the requirements?
	<p>Traditional SRSs (System Requirement Specifications) are put together and agreed for each customer. The PO then sorts the requirements from different customers in the PBL, mainly based on agreed deadlines for each customer. Changes in customers' requirements and deadlines are done all the time; minor changes are handled almost in an agile way resulting in a change in the PBL, but for major ones change request documents are written.</p>
6	Customer involvement
	<p>This set of questions relates to a single customer. If several customers are analyzed, they should be repeated for each. With <i>customer</i> means in this section the end customer – not an in-house Product Owner.</p>
6.1	<p>Is the customer holding the role of the Product Owner?</p>
	<p>Even if fifteen customers exist, they are treated very similar and can be classified as the same type. One exception is the MSP.</p> <p>The customer is not holding the role as PO, but the in-house PO is acting as a customer proxy.</p>
6.2	<p>Is the customer owning and prioritizing the Product Backlog?</p>
	<p>The customer is not owning and prioritizing the Product Backlog, and all requirements are stated in an SRS. The PO owns the PBL.</p>
6.3	<p>Is the customer acting as a project member?</p>
	<p>The customer is not acting as a project member, but the PO is, and somewhat is the CPM. The sales person supporting the PO is not.</p>

APPENDIX CASE STUDY DATA

6.4	Is the customer collocated with the team?
	The customer is not collocated with the team.
6.5	Are the project and all its other customers transparent to this customer?
	<p>Most customers know that they are buying adaptations of a product, and may be aware of other customers acting as references. The development within the Mobical team is not transparent to the customers, and they are not aware of the internal prioritizations between different customers.</p> <p>The development process and prioritizations may be more transparent to the potential MSP, but cooperative plans have not yet been fully drafted.</p>
6.6	How well does the customer know the agile methodology? (0-100%)
	Many customers know some about agile methodologies. Most customers accept and understand that deliveries are made in increments, and that releases of bug-fixes and new features are aligned with sprints. Though, the agile methodology is fairly promoted to the customers, and most customers prefer to work with traditional processes; contracting and SRS followed by one delivery and support. It could be possible that new customers are more open for an agile approach than elder ones.
6.7	Is the customer allowed to attend any meeting?
	The customers are not allowed to attend any of the meetings within the Scrum process, including the sprint demo.
6.8	Is the customer attending the sprint reviews?
	<p>(see above)</p> <p>Separate demonstrations are sometimes arranged for customers.</p>

APPENDIX CASE STUDY DATA

EXPERIENCED AND EXPECTED BENEFITS	
7	Benefits
7.1	Which main benefits is the project experiencing by the agile methods?
	<p>Before Scrum there was a lack of structure in the project. Today Scrum is an important project management tool, bringing structure without making any unnecessary planning, which increases flexibility. The visibility is increased; the prioritizations from sales and customer responsible are visible for everybody in the project, and the awareness of what to do and what everybody is working on has increased. It is also easier for the PO to plan and negotiate with customers about delivery dates. With Scrum, the development process has better abilities to improve, and it is constantly improving. As an example the process for handling defects has improved.</p> <p>Decreased time to market, higher quality and decreased overtime are not significant benefits. Estimation of tasks may have improved, but since the metrics are poor this is uncertain.</p>
7.2	In which areas does the project identify future potential benefits?
	<p>Future potential benefits can generally be identified in all of the areas development, test and customer project management (customer collaboration). A more extensive use of relevant metrics could increase visibility for the team and others, and also increase engagement and commitment by the team.</p>

APPENDIX CASE STUDY DATA

Table A:4 Mobical Investigation protocol – Part 2

1. Delivery Strategy	
A. Regular delivery of software; short iterations of equal length	40
Sprint length is two weeks, but is seldom resulting in increments of shippable functionality. Instead mid-sprint deliveries to specific customers are more important. Neither is it important to finish all tasks within a sprint. Criteria for when a task is complete don't exist, and features are seldom demonstrated on the target systems. The demos tend to trig discussions about what is developed within the teams, which boost communication but not interesting everybody.	
B. Delivering most important features first	100
Yes, features are developed according to the PO's prioritizations.	
2. Agile Software Engineering Techniques	
A. Well-defined coding standards up front	40
Some general agreement about good coding exists, but no formal standards are yet written down. Mobical intends to define standards. Though, this is not experienced as a major limitation.	
B. Pursuing simple design	50
All developers work by "keep it simple", and the design is kept quite simple during the circumstances; legacy code base with complex structure. No general design is defined in design documents, but it is needed and intended to be introduced. Refactoring is done by some and not by others. Code should be cleaned and reviewed, but there is seldom time for this.	
C. Agile documentation handling	75
Traditional SRSs are written for each customer by sales responsible staff. Except these, most internal documentation (specifications etc.) is written on a Wiki or in the project management tool. The project is improving in using the Wiki, which is experienced to work well.	
D. Agile test process	40
Unit tests are written and executed for fundamental layers, but not for the whole system. Time is not invested in writing unit tests for the legacy system, making it not applicable to write tests for the new code. Criteria for when a task is complete don't exist, and function test are not always done within sprint.	

APPENDIX CASE STUDY DATA

E. Daily builds	65
<p>No specific daily builds are made, but the parts of the system are almost always buildable (and build several times per day), and it is seldom hindering not to have daily builds. The tools used today don't support to make automatic daily builds and deployments of the whole system. Sometimes the developed tasks are not even deployed on server each sprint.</p>	
F. Code completed at demonstration	37
<p>Criteria for when a task is complete and for what should be fulfilled at demonstration don't exist. This means that the code isn't as complete as it could be. The demonstrated tasks run on the developers' computers, but are seldom deployed on the target system. In general, everything is not developed and tested within a sprint – there is no time for it, and it is not planned for. A bit of traditional “waterfall” thinking exist, starting with development of a set of features followed by test, then back to developers for bug-fix, back to test, and at last delivery.</p>	
3. Team Capability	
A. Team members with high competence and expertise	55
<p>Most project members are quite senior, but some individual dependencies exist. Some people have unique expertise knowledge (database, server configuration etc.) and sometimes become bottlenecks. Though, everyone should be able to do most tasks if they got time to learn.</p> <p>The two teams are not cross functional, but having specific developers and testers. Though, using specific testers makes developers not test their own code, which is good. The front-end team could be said to be more cross functional than the back-end team, which includes more experts.</p> <p>The lack of cross functionality may be a contributing factor for the team members not to take a common responsibility, but focus on “their own” tasks.</p>	
B. Team members with great motivation and self-discipline	40
<p>Some team members are really dedicated to their work (motivated, self-disciplined and taking initiatives), but some don't show it at all times. Collective responsibility is lacking and work is done more individually than by the team.</p> <p>A reason for the lack of motivation for following Scrum practices may be that the team members don't see the purpose, e.g. time reporting since velocity metrics are not used. There is a common understanding in the project that there is nothing wrong with not fulfilling all tasks within a sprint, which may lead to low engagement.</p>	

APPENDIX CASE STUDY DATA

C. Managers knowledgeable in agile having adaptive management style	55
<p>PO and PM are synched individually, know Scrum and can synch deliveries with sprints. (They are collocated.)</p> <p>CPM is not fully synched with Scrum and sprints, and is not collocated with the others. The poor synch of deliveries and sprints could depend on CPM being responsible for “old” customers with old patterns, but also because Fridays (sprint-delivery days) are not desired for deliveries. Agile aspects are being considered more and more in sale and contracting of projects.</p>	
D. Appropriate technical training to team	75
<p>Knowledge is earned mostly by learning of each other, Internet forums etc. (as in the company in general). There is not always time to study, but the team members are not limited by their knowledge or any lack of training.</p>	
E. Appropriate methodology training to team	40
<p>The methodology knowledge is quite OK for the team members in general, but there may not be enough time to really improve and learn; for people new to their roles (e.g. Scrum Master) and for implementing areas of improvement identified at the process improvement meetings.</p>	
4. Project management process	
A. Following the agile-oriented requirement management process	70
<p>An SRS is agreed with each customer, and these requirements are sorted in the PBL by the PO, CPM and PM. It is not useful to have a separate PBL for each customer since the projects are quite standardized and often short. CPM writes the requirements in traditional “waterfall” ways, but without feedback from developers. Recently a study about translating SRS to backlog tasks was made, and an evaluation of the study is made (but not known by everybody). The study output is not applicable for Mobical without adaptations, and this area can still be improved. A problem today is that the requirements in the end result in too many tasks in the SBL.</p> <p>Placeholders are used in the sprint planning for tasks to be added during the sprint, and about 50% of the time in a sprint is used to develop specified features from the backlog. The rest is used on placeholder activities and on unforeseen activities, making it impossible to finish all tasks in a sprint. These unforeseen activities are based on bad estimations, e.g. more support than planned, sale support, time for task-switching, and to low estimations generally.</p>	

APPENDIX CASE STUDY DATA

B. Following the agile-oriented project management process	45
<p>Self-organization of the teams works quite well at the sprint planning. Then placeholder activities, support errands and sale support errands (not planned for) are added, and it sometimes is unclear who is prioritizing the tasks (when CPM adds tasks by calling a team member). The teams are not allowed to work undisturbed and independently, and tend to not being able to self-reorganize. This makes the teams not taking a commit seriously, and they don't take common responsibility for the tasks. The teams are not enough self-managing, and an "end-of-sprint spirit" doesn't exist.</p> <p>The PM identifies metrics and uses them, but mostly for himself. Generally there is too little time to improve the agile process.</p>	
C. Good progress tracking mechanism	40
<p>The used project management tool is not bad, neither optimal. The ones that need to see progress understands the tool, but outside stakeholders probably don't. The follow up of metrics – velocity, feature turnaround speed (from story in PBL to on the market) etc. – is poor, and team members are not motivated to be disciplined in reporting.</p>	
D. Strong communication focus with daily face-to-face meetings	75
<p>Daily Scrum meetings are held. The customer is not communicated with on a daily basis, but the PO is. The CPM, located in another city, is not communicated with every day, and rarely face-to-face.</p>	
E. Honoring regular working schedule	80
<p>Some peaks cause overtime at deliveries, but not often longer than a week at a time. Low pressure on finishing all tasks in a sprint may also keep down overtime.</p>	
5. Team environment	
A. Co-location of the whole team	55
<p>The CPM and the seller are located in other cities, and the PO has recently moved from being collocated with the team to a separate sales office on a different floor in the same building. This gives the PO better opportunities to focus on the sales activities (the customers), but may slightly complicate communication with the team.</p> <p>It works well for the PO with the CPM not collocated, but it had made communication easier for the team, especially close to a delivery.</p>	

APPENDIX CASE STUDY DATA

B. Coherent, self-organizing teamwork	45
Self-organizing teamwork at sprint planning, but not more. The teams are disturbed by added tasks and support errands. (As described in 4.B)	
C. Projects with small teams	100
5 people in each.	
D. Projects with no multiple independent teams	75
The tasks for the two teams are rarely independent, but since the teams are collocated, communication is easy.	
6. Customer involvement	
A. Good customer relationship	88
With regard to support (external customers), relations are good. Relation with the PO is very good.	
B. Strong customer commitment and presence	63
PO commitment and presence is very good, but CPM is not as good. He is present by his requirements.	
C. Customer having full authority.	100
PO (and CPM and PM) have full authority.	
D. Customers trained in the agile process	55
The end customers are not, but the PO is. The CPM is not, but the PM is an expert.	
E. Use of target-cost contracts to share risk	N/A
Target-cost contracts could maybe be applicable in future projects, but most customers today are much focused on a fixed price, and keeping their budget. The target-cost approach is also hard to apply when selling and delivering a standardized product according to the pre-specified requirements.	

APPENDIX CASE STUDY DATA

Case B – Audio Control

Investigation date: 2008-11-14
 Interviewees: Team Leader (Scrum Master, Product Owner and developer)
 Developer

Table A:5 Audio Control Investigation protocol – Part 1

	BACKGROUND
1	General about the company
1.1	Are there any limitations in publishing the collected data?
	No
1.2	Name of the company?
	Tactel AB
1.3	Founded?
	Mid 1990s
1.4	What types of products does the company develop?
	Tactel is a developer of mobile applications, providing solutions and consulting services to many of the world’s major operators and handset vendors
1.5	Who are the major customers for the company?
	(see above)
1.6	How many employees?
	About 350
1.7	Multinational?
	Offices in Sweden, USA, Ukraine. Customers globally.
1.8	Does the company have experience of agile methodologies in several projects?
	In some.
2	General about the project
2.1	Name of project
	Audio Control

APPENDIX CASE STUDY DATA

2.2	What types of products does the project develop?
	<p>Audio Control (AC) develops and maintains a software module in Sony Ericsson's mobile phones. AC is run as a sub-project in Sony Ericsson's organization, and is contracted on an outsourced basis.</p> <p>Sony Ericsson is the only customer, not buying a product but rather hours; a group of engineers during a specified period of time. The development consists of about 90% maintenance and support and 10% new development.</p> <p>Tactel has handled AC as an outsourced project for more than five years.</p>
2.3	How could the project be described in the company's organization?
	<p>As Mobical.</p> <p>Audio control recently was divided in two teams; back-end and front-end. This study focuses on the front-end team.</p>
2.4	Who are the major customers for the project?
	See above.
2.5	Is the project using any project management tool?
	A Wiki is used for the SBL. This is a very simple tool, and the implementation of it was easy. It is not limiting, and the visibility is good enough. Though, there is poor support for metrics.
3	Agile project specifics
3.1	Which agile method or methods are in use?
	Scrum
3.2	For how long has the project been managed with agile methodologies?
	3 months

APPENDIX CASE STUDY DATA

3.3	<p>To what extent is the project following the agile practices “by the book”?</p> <p>The customer organization has a task-management tool, where all tasks from different customer departments are prioritized. This tool serves as a PBL, and no separate PBL exist. This tool is hereafter referred to as the PBL.</p> <p>No actual PO exists; the customer could have had this role, and being active in prioritizing tasks at the sprint-planning meeting. The one communicating and reporting to the customer is the Team Leader, and as he is responsible for getting the right things done – putting the most important tasks in the SBL – he is considered as PO in this study.</p> <p>At the sprint planning, the AC team analyzes the tasks in the “PBL”, sometimes re-prioritizes them, and puts the highest prioritized tasks together in the SBL.</p> <p>Tasks are often added during the sprints, a matter that could be solved by placeholders. AC has tried different ways to work with this, but is still figuring out which is the best.</p> <p>Deliveries are not synched with sprints, but are made as soon as PBL-task are finished. As a consequence, end-of-sprint demos are not exercised.</p> <p>Though, there could perhaps be a point in demonstrating new developed features, but this is only a small part of the total work; today less than 10%.</p> <p>Most tasks are finished within a sprint, but it is not necessary to finish all, and an end-of-sprint spirit doesn't exist.</p>
4	Group organization
4.1	Total number of people in the project?
	5 developers
4.2	Agile Roles in the project?
	SC (and PO) is held by the Team Leader
4.3	Support Roles in the project?
	<p>One architect and one technical coordinator support the both AC teams, and are not part of the Scrum teams.</p> <p>All deliveries are made to an internal test department, collocated with the teams. They support both development teams, and also work outside the agile process.</p> <p>Business relations between Tactel and the customer are handled by managers outside the project, and consider several other projects besides AC.</p>
4.4	Who in the project are responsible for customer relations?
	<p>The Team Leader is primarily responsible for customer relations, and attends several meetings on-site at the customer each week.</p> <p>On business levels, senior managers are also responsible.</p>

APPENDIX CASE STUDY DATA

CUSTOMER CHARACTERISTICS	
5	General customer characteristics
5.1	<p>Does the project have one or multiple customers?</p> <ul style="list-style-type: none"> • How many? • Are there differences between them in size and importance?
	<p>AC has one main customer, but 3-4 different departments in the customer's organization are receivers of deliveries – and could then be seen as different customers. Though, they are all of the same type and treated alike, and sort their requirements in a common "PBL". They are hereafter referred to as one customer.</p>
5.2	<p>Does the project have a separate backlog for each customer?</p> <ul style="list-style-type: none"> • If not, how does the project handle the requirements?
	<p>In the PBL-like task management tool, tasks from different departments are prioritized by the main customer, and in this meaning only one PBL exists.</p>
6	Customer involvement
	<p>This set of questions relates to a single customer. If several customers are analyzed, they should be repeated for each. With <i>customer</i> means in this section the end customer – not an in-house Product Owner.</p>
6.1	<p>Is the customer holding the role of the Product Owner?</p>
	<p>No. The Team Leader is responsible for final prioritizations, but still the customer outlines the prioritizations in general.</p>
6.2	<p>Is the customer owning and prioritizing the Product Backlog?</p>
	<p>Yes, almost.</p>
6.3	<p>Is the customer acting as a project member?</p>
	<p>No.</p>
6.4	<p>Is the customer collocated with the team?</p>
	<p>No.</p>
6.5	<p>Are the project and all its other customers transparent to this customer?</p>
	<p>The project is almost totally transparent, and could be completely visible for the customer.</p>
6.6	<p>How well does the customer know the agile methodology? (0-100%)</p>
	<p>Scrum is implemented in several projects on lower levels within the customer organization. The managers respond positively to Scrum, but are not yet taking part of the process themselves, e.g. by adapting reporting structures and deliveries to Scrum.</p>

APPENDIX CASE STUDY DATA

6.7	Is the customer allowed to attend any meeting?
	Yes, but they are not. As long as the work runs as it should there is no need. Both the customer and tem are quite satisfied as it is; they don't spend time on unnecessary meetings.
6.8	Is the customer attending the sprint reviews?
	No, but perhaps the project could benefit from engaging the customer more, by making them part of the final prioritizations.
EXPERIENCED AND EXPECTED BENEFITS	
7	Benefits
7.1	Which main benefits is the project experiencing by the agile methods?
	<p>The short sprints make it possible to start from a blank sheet every other week; to evaluate the past and make a new planning. It feels better if all tasks are finished within a sprint, but regardless if they are or not, the team experiences the re-start very motivating.</p> <p>The communication climate has improved with Scrum; everyone knows what the others are working on, they are more interested in each other's work and get faster help and answers to their questions.</p> <p>The progress is more visible for the Team Leader by the daily meetings and updates. The daily follow-ups also spur the team members to get things done.</p> <p>Prioritizations and structure are clearer, making the work process more efficient, and today there is time to build code with higher quality.</p>
7.2	In which areas does the project identify future potential benefits?
	<p>The efficiency could be increased even more with better metrics and follow-ups of them, but could also increase time spent on reporting.</p> <p>If the customer adapted their processes to Scrum, prioritizations could be clearer and releases better aligned with sprints.</p>

APPENDIX CASE STUDY DATA

Table A:6 Audio Control Investigation protocol – Part 2

1. Delivery Strategy	
A. Regular delivery of software; short iterations of equal length	65
Sprints are two weeks, but deliveries are not aligned with them. Support and maintenance tasks are delivered when they are finished, but new developed features could be better aligned with the sprints.	
B. Delivering most important features first	90
The customer and the team not always prioritize in the same way, but the highest prioritized tasks (according to the customer) are always delivered on time. Sometimes the team may start with longer lower ranked tasks before shorter higher ranked to get a better flow in the development process. Though, the sprint-planning is followed to 100%.	
2. Agile Software Engineering Techniques	
A. Well-defined coding standards up front	75
The customer's standards are followed quite well.	
B. Pursuing simple design	60
With Scrum the team has started reviewing and refactoring the code, but further improvements are possible. It is easier to apply simple-design principles to new than to legacy code, and it is also a requirement from the customer not to take the risk to change working code.	
C. Agile documentation handling	75
The customer requires traditional documentation to some extent (project specifications, interface descriptions etc.), but the design and documentation of the code is done when it is developed. The SBL is documented on a Wiki.	
D. Agile test process	50
Test-driven development (TDD) is considered but not implemented, and unit tests are not used. Developers test their code before committing, and then deliver to the test department who runs automatic test series. It would be hard to write unit tests to legacy code, but perhaps quality could increase a bit with unit tests.	
E. Daily builds	90
It is time-consuming to update all code and make a build, and the 3-4 different sub-projects build about twice a week. This is considered enough.	

APPENDIX CASE STUDY DATA

F. Code completed at demonstration	85
Demonstrations don't exist, but when code is delivered it is completed. Criteria for when a task is done exist, and are followed. <i>Done</i> means that the development is finished, the code is reviewed, documented, built and sent to test.	
3. Team Capability	
A. Team members with high competence and expertise	85
All developers are quite senior. Some are more experienced in certain areas, but are not hindering the development, and tasks within the expertise areas are divided between all developers. There is a common responsibility for the work, and the team is quite cross functional.	
B. Team members with great motivation and self-discipline	95
All team members are interested in Scrum and learn quickly. They follow the guidelines and have a positive attitude. Making the Scrum process more complex by imposing better estimations, time-reporting etc. could decrease motivation – today Scrum is simple.	
C. Managers knowledgeable in agile having adaptive management style	50
The managers within the customer organization respond positively to Scrum, but are not yet taking part of the process themselves, e.g. by adapting reporting structures and deliveries to Scrum.	
D. Appropriate technical training to team	70
OK. They learn by each other, search knowledge at Internet forums etc.	
E. Appropriate methodology training to team	75
This far all team members know Scrum well enough, and all have been trained. There have not yet been a need for strategies for future methodology improvement, but the lack of strategies keeps the score down.	

APPENDIX CASE STUDY DATA

4. Project management process	
A. Following the agile-oriented requirement management process	95
Yes, "PBL" tasks are transferred to SBL.	
B. Following the agile-oriented project management process	95
Teams are self-organizing.	
C. Good progress tracking mechanism	75
The progress is visible on the Wiki page (SBL), but no use of burn-down charts or velocity. The good communication climate makes progress clear for at least the Team Leader, and there is no critical need for a better PM tool measuring progress.	
D. Strong communication focus with daily face-to-face meetings	100
Yes.	
E. Honoring regular working schedule	95
Yes, almost no overtime.	
5. Team environment	
A. Co-location of the whole team	75
Yes, but Team Leader a little distanced (even if in the same room).	
B. Coherent, self-organizing teamwork	90
Yes, as described above.	
C. Projects with small teams	100
Five in each team.	
D. Projects with no multiple independent teams	75
Both AC teams develop the same code base, and dependencies exist. They are minimized by the division in front-end and back-end, but even if both teams are located in the same office space, they are not communicating very much. Communication between the teams can improve.	

APPENDIX CASE STUDY DATA

6. Customer involvement	
A. Good customer relationship	80
AC makes a good job and the customer is satisfied. The customer understands and sees progress in the work, and the relation is good. (Since the customer is classified as Partial Participant, this driver relates to the end customer.)	
B. Strong customer commitment and presence	100
The customer is not present at the development site very often, but the Team Leader (acting as PO) is collocated with the team, and also acting as Scrum Master. The team meets the customer at general assemblies once or twice a year. (Since the customer is classified as Partial Participant, this driver relates to the in-house PO.)	
C. Customer having full authority.	100
The customer prioritizes the "PBL", but it is edited by the team, with the Team Leader as ultimately responsible. (Since the customer is classified as Partial Participant, this driver relates to the in-house PO.)	
D. Customers trained in the agile process	50
The customer is not, but not hindering. The Team Leader is. (Since the customer is classified as Partial Participant, this driver relates to the end customer.)	
E. Use of target-cost contracts to share risk	N/A
Not applicable since the customer is buying hours and not a product.	

APPENDIX CASE STUDY DATA

APPENDIX CASE STUDY DATA

Case C – Cascades and Kastor

Investigation date: 2008-11-14 and 2008-11-24
 Interviewees: Product Owner
 Scrum Master

Table A:7 Cascades and Kastor Investigation protocol – Part 1

BACKGROUND	
1	General about the company
1.1	Are there any limitations in publishing the collected data?
	No
1.2	Name of the company?
	TAT, The Astonishing Tribe, AB
1.3	Founded?
	2002
1.4	What types of products does the company develop?
	<p>TAT is a Swedish software technology and design company offering products and services that differentiate and enhance the user experience of portable devices.</p> <p>TAT's mission: "TAT is a company where design and technology are two sides of the same coin. It requires a culture and a company philosophy that creates a working environment that encourages change, curiosity and exploration."</p> <p>TAT offers a comprehensive suite of products and services for the creation of advanced mobile user interfaces. The TAT market offering is divided into two main areas, products and services, and the main offerings are as follows.</p> <ul style="list-style-type: none"> • <i>TAT Cascades</i> is a UI framework for the production of advanced user interfaces. • <i>TAT Motion Lab</i> is an XML development environment for TAT Cascades. • <i>TAT Kastor</i> is a powerful UI rendering platform. • TAT also offers complete UI design projects from start to final implementation.
1.5	Who are the major customers for the company?
	TAT works with 4 of the 6 leading OEMs (Original Equipment Manufacturers) in the mobile device space today. Publicly announced clients include Sony Ericsson, Motorola, S60, Samsung, Vodafone and Orange.

APPENDIX CASE STUDY DATA

1.6	How many employees?
	Since the company was founded in 2002 it has continuously grown, and employs in November 2008 about 150 people.
1.7	Multinational?
	TAT is headquartered in Malmö, Sweden, and has local offices in Korea and USA.
1.8	Does the company have experience of agile methodologies in several projects?
	Agile and iterative development methods are used in several projects within TAT, but all are not working specifically according to Scrum.
2	General about the project
2.1	Name of project
	The studied project is the development of Cascades and Kastor. The development of both products is much related since the main product Cascades is built on Kastor, they have the same PO, and they have about the same customers. Therefore they are in this study treated as one project, where different Scrum teams work with the different products.
2.2	What types of products does the project develop?
	As described above. The development of a release normally takes three months. After two months is a pre-release made, which is tested and stabilized, and after three months the new version of the product is released.
2.3	How could the project be described in the company's organization?
	Development department manager 3 Product owners 4 Scrum teams, of which two are Cascades and Kastor. A re-organization is about to take place within the development department.
2.4	Who are the major customers for the project?
	As described above. The studied project develops products are sold by license.
2,5	Is the project using any project management tool?
	The web-based tool JIRA is used for product- and sprint-backlog handling, metrics etc. It works OK, but measures are not optimal. The burn-down charts was more visible when excel was used. Velocity is not measured, but high prioritized to implement.

APPENDIX CASE STUDY DATA

3	Agile project specifics
3.1	Which agile method or methods are in use?
	Scrum
3.2	For how long has the project been managed with agile methodologies?
	Scrum has been implemented for two years. Before Scrum the project worked in an unspecified agile way.
3.3	To what extent is the project following the agile practices “by the book”?
	<p>Sprint-length has always been two weeks, but it is just decided to change to three weeks.</p> <p>All tasks assigned for a sprint are seldom finished; often only about two thirds are. Since this is the common way of working, the tasks are prioritized in the SBL.</p> <p>Features often take more than two weeks to develop, and as a consequence not much attention is paid to demos at the end of each sprint. The definition of when a task is “done” (code review, documentation, test etc.) is vague, with the result that the PO not always knows the actual status of the development. Defining “done” and paying more attention to sprint demos could result in better structure and increased focus in the project.</p> <p>Scrum doesn’t describe any test process well, and feature test is mainly handled outside the agile process.</p> <p>TAT is always trying to improve their processes, and is already well advanced according to an external Scrum consultant.</p>
4	Group organization
4.1	Total number of people in the project?
	<p>15-20, at the moment with about ten in Cascades and 6 in Kastor.</p> <p>Temporary <i>Feature Teams</i> are sometimes put together when the development of an important task or feature concerns Cascades, Kastor and other related teams. The Feature Teams contain members from several Scrum teams. It is discussed to create permanent Feature Teams, handling cross-product development, not least because the team sizes are a bit too large.</p> <p>Function and feature tests are performed at a separate test department, and is not involved in the Scrum process.</p>
4.2	Agile Roles in the project?
	<p>One PO.</p> <p>Two SMs, which in general not are part of the teams.</p>

APPENDIX CASE STUDY DATA

4.3	Support Roles in the project?
	No support roles.
4.4	Who in the project are responsible for customer relations?
	Sales and marketing organization with Key Account Manager (KAM) and Technical Account Manager (TAM), and the PO.
CUSTOMER CHARACTERISTICS	
5	General customer characteristics
5.1	Does the project have one or multiple customers? <ul style="list-style-type: none"> • How many? • Are there differences between them in size and importance?
	<p>The project has about five key customers, five additional major customers and some minor ones. The key customers are mainly influencing the development by putting requirements on future releases. The additional major customers are also requiring features to develop and bugs to fix, but are not prioritized as high. The minor customers' requirements have lowest priority to include in future releases.</p> <p>Further requirements and features are added to meet requirements from potential new customers. This is done by having a dialogue with them, by being updated in new technology and by following the <i>leads</i> – the leading developers – in the business.</p>
5.2	Does the project have a separate backlog for each customer? <ul style="list-style-type: none"> • If not, how does the project handle the requirements?
	No. They buy a product and get new releases. One release about every three months.
6	Customer involvement
	This set of questions relates to a single customer. If several customers are analyzed, they should be repeated for each. With <i>customer</i> means in this section the end customer – not an in-house Product Owner.
6.1	Is the customer holding the role of the Product Owner?
	None of the customers is.
6.2	Is the customer owning and prioritizing the Product Backlog?
	No, but the PO is.
6.3	Is the customer acting as a project member?
	No, but the PO is.
6.4	Is the customer collocated with the team?
	No. The PO, though, is located in an own office at the same floor as the team.

APPENDIX CASE STUDY DATA

6.5	Are the project and all its other customers transparent to this customer?
	No. Some customers are public announced, but none of the customers know the prioritization between them.
6.6	How well does the customer know the agile methodology? (0-100%)
	The end customer is to some extent approached in an agile way when it comes to discuss ideas, but in general the customers are pleased with stable deliveries every three months.
6.7	Is the customer allowed to attend any meeting?
	No.
6.8	Is the customer attending the sprint reviews?
	No.
EXPERIENCED AND EXPECTED BENEFITS	
7	Benefits
7.1	Which main benefits is the project experiencing by the agile methods?
	<p>The introduction of Scrum brought a common vocabulary to the project, which have resulted in a better structure. It helps the team to focus on the right things, supports regular follow-ups and re-prioritizations, and enables continual risk management.</p> <p>Time to market is generally not reduced by Scrum, but the products are easier adapted to end customers. The improved customer-orientation has resulted in new business.</p> <p>Other benefits with Scrum are simplicity and communication – people sit down to talk and solve problems. Now people are also allowed to work more independent and experience that they have more influence.</p>
7.2	In which areas does the project identify future potential benefits?
	<p>Areas to improve to gain future benefits are:</p> <ul style="list-style-type: none"> • Finding the right format for a common PBL for all products. • Better organize in Feature Teams, and co-location of them. • Better coordination between the teams. • More extensive testing on test-server. • Better use of metrics. • Defining <i>done</i>. • Reduce waste in general. <p>Further process improvements could result in higher development speed and increased quality, but not necessarily due to Scrum.</p>

APPENDIX CASE STUDY DATA

Table A:8 Cascades and Kastor Investigation protocol – Part 2

1. Delivery Strategy	
A. Regular delivery of software; short iterations of equal length	50
Two-week sprints have been practiced, and is about to be changed to three, but all tasks are seldom finished within a sprint. Some tasks are added outside the backlogs, as urgent bugfixes. Vague definition of “done”, features sometimes take more than two weeks to develop, and not much attention is paid to demos.	
B. Delivering most important features first	100
Features are always developed according to the prioritizations of the PO.	
2. Agile Software Engineering Techniques	
A. Well-defined coding standards up front	65
A good hygienic level is kept in most engineering-technique areas. Not too much attention is paid, neither are they ignored; they support the development well.	
B. Pursuing simple design	65
Refactoring is done when needed. Some complex features, which have been re-built with added functions, could be better refactored.	
C. Agile documentation handling	75
Design documentation is not extensive, and documented on a Wiki.	
D. Agile test process	80
TDD is not exercised but automatic tests are written and run for most features.	
E. Daily builds	95
All check-ins are built and tested automatically on a continuous integration server. The build results are visible to all team members on a large screen in the office space.	
F. Code completed at demonstration	35
Poor attention is paid to demos, and the definition of “done” is vague. This is partly because of dependencies between the Scrum teams, and that all features cannot be finished until another team has finished related tasks.	

APPENDIX CASE STUDY DATA

3. Team Capability	
A. Team members with high competence and expertise	75
<p>The general technical knowledge is good, and the developers quite senior. Some people naturally are experts in certain areas, but the team members try to share their knowledge with each other. It is not experienced that the individual expertise leads to bottle necks, but certainly is a risk. The different experts with their different roles complement each other.</p> <p>The teams are in general quite self-managing, and tasks are not assigned to a developer by a project manager or similar. The self-managing approach may not always be the most efficient, at least not in the short run.</p> <p>The Feature teams have a better ability to self-manage and focus on their tasks.</p>	
B. Team members with great motivation and self-discipline	90
<p>The teams are strongly motivated and take responsibility. Most project members have a positive attitude and put effort in the agile practices. A minority is indifferent.</p>	
C. Managers knowledgeable in agile having adaptive management style	80
<p>Managers in direct contact with the project (Product management and managers in the development organization) are knowledgeable in agile methods. Other managers are not, but this is not hindering the project. The PO, though, has to adapt his information and “translate” it when reporting to senior management.</p>	
D. Appropriate technical training to team	85
<p>The technical training is good and people are learning by each other, attending conferences etc.</p>	
E. Appropriate methodology training to team	85
<p>Some driving forces are more interested and train themselves, people are sent to conferences, read books, teach each other. The team recently got inspired to improve their Scrum process by a “Scrum day”, where a lecture about Scrum was held.</p>	

APPENDIX CASE STUDY DATA

4. Project management process	
A. Following the agile-oriented requirement management process	100
The roadmap and a detailed roadmap (both owned by the PO) is translated to tasks in the PBL. One PBL exist today for Kastor and one for Cascades, but one common PBL is planned for together with other related projects.	
B. Following the agile-oriented project management process	75
The PO prioritizes the backlog, and tasks are put in the SBL at the sprint planning meeting. The tasks are then distributed by the team members themselves.	
C. Good progress tracking mechanism	50
JIRA works OK, but measures are not optimal. The burn-down charts was more visible when excel was used. Velocity is not measured, but high prioritized to implement. The progress tracking is not optimal (low visibility), but this is not critical since all features within a task seldom are developed anyway.	
D. Strong communication focus with daily face-to-face meetings	85
The communication climate is good, and the teams meet for daily Scrums, and are collocated. The PO attends about 1/3 of the meetings.	
E. Honoring regular working schedule	85
Scrum doesn't significantly affect the amount of overtime. There are minor peaks prior to pre-releases and releases, but not much overtime is worked.	
5. Team environment	
A. Co-location of the whole team	85
Each team is collocated, and all teams are located on the same floor. The PO has his own office on the same floor close to the teams.	
B. Coherent, self-organizing teamwork	75
The teams are in general quite self-managing, and tasks are not assigned to a developer by a project manager or similar. The self-managing approach may not always be the most efficient, at least not in the short run. The Feature teams have a better ability to self-manage and focus on their tasks. The teams get better and better in self-organizing. The SMs try to find the right level of what to control and what to leave to the teams.	

APPENDIX CASE STUDY DATA

C. Projects with small teams	60
One of the teams is a bit too large, and ten people make it harder to keep a common focus on the tasks and the development. The other team has a good team size.	
D. Projects with no multiple independent teams	40
There are many dependencies between the teams, concerning approximately 20% of the development. The teams are located at the same floor, and Feature Teams are sometimes put together for advanced cross-product features. Many dependencies between the different teams often make it hard to finish tasks, and all features are not done at sprint end. This is one of the areas that TAT would like to improve. One step is to align the different roadmaps to one single PBL, and another to better work with Feature Teams.	
6. Customer involvement	
A. Good customer relationship	80
The PO prioritizes the backlogs, and attends several of the meetings. He is also familiar with the products on a feature level.	
B. Strong customer commitment and presence	70
Located on the same floor, attends 1/3 of the meetings.	
C. Customer having full authority.	100
Yes, since the PO's prioritizations always are followed.	
D. Customers trained in the agile process	85
Yes, the PO knows agile methods well.	
E. Use of target-cost contracts to share risk	N/A
Not applicable for Kastor and Cascades, since customers are buying a product.	