Institutionen för datavetenskap
Department of Computer and Information Science

Master Thesis

Quality assurance with TL 9000 in agile software development of set-top boxes
The case of Motorola and the use of Scrum

by

Kristofer Gustafsson and Johan Jacobsson

LIU-IDA/LITH-EX-A--09/011--SE

2009-03-25
Master Thesis

Quality assurance with TL 9000 in agile software development of set-top boxes
The case of Motorola and the use of Scrum

by

Kristofer Gustafsson
Johan Jacobsson

LIU-IDA/LITH-EX-A--09/011--SE

2009-03-25

Supervisor: Håkan Markör, Motorola
Mattias Hedlund, Motorola

Examiner: Vivian Vimarlund, IDA
In today’s fast-paced world, there is a constant demand for better and more efficient ways of doing business. Motorola in Linköping are using the agile development framework, Scrum in their software development. A certain level of quality must also be assured of the delivered goods and services. Is it possible to use Scrum and still meet the quality requirements?

This Master Thesis is performed to investigate if it is possible to achieve a quality certificate from TL 9000, the telecom industry extension of ISO 9000, when using the agile development framework Scrum. The investigation consists of interviews and observations at Motorola, Linköping, along with literature studies about quality systems and agile development.

The conclusion is that it is possible to meet the TL 9000 requirements when using Scrum, under the condition that some additional processes are performed and that other parts of the organization also fulfills the remaining requirements. This is needed since there are requirements that are out of scope for the Scrum framework. Examples of the suggested additions are to follow the Scrum framework more strictly and to adopt a more specific definition of done.
Abstract

In today’s fast-paced world, there is a constant demand for better and more efficient ways of doing business. Motorola in Linköping are using the agile development framework Scrum in their software development. A certain level of quality must also be assured of the delivered goods and services. Is it possible to use Scrum and still meet the quality requirements?

This Master Thesis is performed to investigate if it is possible to achieve a quality certificate from TL 9000, the telecom industry extension of ISO 9000, when using the agile development framework Scrum. The investigation consists of interviews and observations at Motorola, Linköping, along with literature studies about quality systems and agile development.

The conclusion is that it is possible to meet the TL 9000 requirements when using Scrum, under the condition that some additional processes are performed and that other parts of the organization also fulfills the remaining requirements. This is needed since there are requirements that are out of scope for the Scrum framework. Examples of the suggested additions are to follow the Scrum framework more strictly and to adopt a more specific definition of done.
Acknowledgements

After five intense months of reading, discussing, and writing, this Master Thesis concludes our Master of Science at Linköping University. Without the help and support from a number of people, we doubt that this thesis would have been written.

First, we would like to thank our supervisors, examiner, and opponent. Håkan Markör and Mattias Hedlund at Motorola kindly offered us this interesting opportunity and continued to inspire, facilitate, and question our work. Vivian Vimarlund, Ph.D., at Linköping University contributed with invaluable feedback, suggestions, and advice throughout the project. Andreas “Goken” Anderljung read the report several times, and helped us by pointing out and suggest improvements for the weak parts. Thank you!

We would also like to thank everyone else that has helped us in any way. Specifically we would like to mention a few people.

Our friends and colleagues studying Master of Science in Information Technology at Linköping University helped us understand how a team works, why a whiteboard is a necessary tool and that “fika” is a great thing. The time we spent together was not just fun, we actually gained invaluable experience.

Susanne Ejdfors Eriksson welcomed us to share her room the first weeks and introduced us to the company. Niklas Storm generously shared his expertise concerning quality management and TL 9000 by helping us interpret the Requirements Handbook. Henrik Gidmark offered us rewarding discussions about group dynamics, leadership, and organizational issues sharing his experience from the corporate world. John Edeus always had time for a chat and a joke, and open heartedly shared his wisdom and real life experience. Mikael Adlers gave us a better understanding of what Scrum is and how it is used at Motorola. Ivy Pascoe and Jos Koster invited us to attend a TL 9000-audit that clarified the connection between theory and practice to us. Carl Ljungström kept us company, especially when we were busy, and helped us balance between working harder and relaxing. The interviewees, you know who you are, gave us unique insights into the organization that would have been impossible to gain otherwise. Everyone at Motorola Linköping helped us with a warm and welcoming attitude.

Finally, we would like to thank our families and friends. You encouraged and supported us. We owe you everything!

Based on the feedback and insights offered by reviewers, we have attempted to address the issues raised. However, any errors, oversights, or omissions remaining are our sole responsibility.

Linköping, February 2009

Kristofer Gustafsson                Johan Jacobsson
# Table of contents

1 Introduction ........................................................................................................... 1  
1.1 The Case ............................................................................................................1  
1.2 Aim ................................................................................................................... 2  
1.3 Problem definition ........................................................................................... 2  
1.4 Limitations of Scope ........................................................................................ 2  

2 Method ................................................................................................................... 3  
2.1 Methodical orientation .................................................................................... 3  
2.2 Qualitative or Quantitative research strategy ................................................. 3  
2.3 Triangulation ................................................................................................... 4  
2.4 Case study ........................................................................................................ 4  
2.4.1 Reliability and Validity............................................................................ 5  
2.4.2 Case Study Protocol ................................................................................. 7  
2.5 Data gathering techniques............................................................................... 8  
2.5.1 Literature study ....................................................................................... 8  
2.5.2 Interviews................................................................................................. 9  
2.6 Work process.................................................................................................. 10  
2.7 Methods used for this study ...........................................................................11  
2.7.1 Interviews................................................................................................12  

3 Quality ................................................................................................................. 15  
3.1 Quality Theory ................................................................................................15  
3.1.1 Quality Management Systems ...................................................................15  
3.1.2 Cornerstones in Total Quality Management .........................................16  
3.2 Quality Tools .................................................................................................. 18  
3.2.1 The ISO 9000 family.............................................................................. 18  
3.2.2 Quality in Software development ..........................................................19  
3.2.3 Telecommunication Quality Standard - TL 9000 .................................19  
3.2.4 TL 9000 Requirements ..........................................................................20  

4 Scrum ................................................................................................................... 25  
4.1 Agile development ......................................................................................... 25  
4.2 Introduction to Scrum ................................................................................... 26  
4.3 What is Scrum? .............................................................................................. 27  
4.4 Scrum roles .................................................................................................... 29  
4.4.1 The Scrum Team .................................................................................... 29  
4.4.2 The Scrum Master.................................................................................. 29  
4.4.3 The Product Owner ................................................................................30  
4.5 Practices ......................................................................................................... 30  
4.5.1 Sprint ...................................................................................................... 30  
4.5.2 Self-organization .................................................................................... 31  
4.5.3 Sprint Planning Meeting ....................................................................... 32  
4.5.4 Daily Scrum............................................................................................ 32  
4.5.5 Sprint Review.......................................................................................... 33  
4.5.6 Sprint Retrospective .......................................................................... 33  
4.5.7 Working environment ........................................................................... 33  
4.6 Artifacts .......................................................................................................... 34  
4.6.1 Product Backlog ................................................................................... 34
4.6.2 Sprint Backlog ................................................................. 34
4.6.3 Release Backlog ............................................................. 35
4.6.4 Burn-down and Burn-down chart ................................... 35
4.7 Using Scrum in general ......................................................... 35
4.7.1 Suitable circumstances .................................................. 36
4.7.2 Advantages and disadvantages with Scrum ..................... 36
4.7.3 Scaling Scrum ............................................................... 37
4.7.4 Disagreement ............................................................... 37
4.8 Combining agile development and ISO 9001 ......................... 37

5 Result ........................................................................................................... 41
5.1 Scrum roles .......................................................................................... 41
5.1.1 The Scrum Team .............................................................................. 41
5.1.2 The Scrum Master ........................................................................... 42
5.1.3 The Product Owner .......................................................................... 42
5.1.4 Project Manager ............................................................................... 43
5.2 Practices in Scrum ................................................................................. 43
5.2.1 Sprint ............................................................................................ 43
5.2.2 Sprint Planning Meeting ................................................................. 43
5.2.3 Daily Scrum .................................................................................... 45
5.2.4 Scrum of Scrum ............................................................................. 45
5.2.5 Demonstration ................................................................................ 45
5.2.6 Retrospective .................................................................................. 46
5.2.7 Working environment ...................................................................... 46
5.2.8 Self-organizing ............................................................................... 46
5.3 Artifacts used in Scrum ....................................................................... 47
5.3.1 Backlog ........................................................................................ 47
5.3.2 Burn-down chart ............................................................................ 48
5.4 Documentation ...................................................................................... 48

6 Discussion and Analysis ............................................................................ 49
6.1 Scrum and quality at Motorola ......................................................... 49
6.1.1 Scrum and Total Quality Management ........................................ 50
6.2 Differences between the Scrum framework and Scrum at Motorola 51
6.2.1 Common Backlog ......................................................................... 51
6.2.2 Product Owner-team ...................................................................... 51
6.2.3 Sprint Planning .............................................................................. 52
6.2.4 During the Sprint ............................................................................ 52
6.2.5 Roles and responsibilities ............................................................... 52
6.2.6 Scrum of Scrum ............................................................................. 53
6.2.7 At the end of the Sprint ................................................................. 53
6.2.8 Customer collaboration and Release Management .................... 54
6.2.9 Conclusions about the use of Scrum at Motorola ....................... 54
6.3 Combining ISO 9000 and Scrum ......................................................... 55
6.4 Scrum at Motorola compared to the requirements of TL 9000 ........ 56
6.4.1 Summary of the comparison ......................................................... 57
6.4.2 Comparison of the TL 9000 requirements and Scrum at Motorola 59
6.5 Changes needed to be compliant with TL 9000 ............................ 73
6.5.1 Definition of Done ......................................................................... 73
6.5.2 Sprint Report .................................................................................. 74
6.5.3 Product Backlog ............................................................................. 74
6.5.4 Release .......................................................................................... 75
List of Tables and Figures

Figure 1. Reliability and validity using arrows on a dartboard................................. 6
Figure 2. Thesis solution path .................................................................................. 11
Figure 3. Bad and good circle of motivation affection of the result............................17
Figure 4. The TL 9000 Layer model. ......................................................................20
Figure 5. Manifesto for Agile Software Development from Agile Alliance.............. 25
Figure 6. Scrum as a framework ............................................................................ 28
Figure 7. Project trade-off triangle ......................................................................... 31
Figure 8. Relationship to the Product Backlog ....................................................... 34
Figure 9. A typical Burn-down-chart ................................................................. 35
Figure 10. How Scrum works ............................................................................... 36
“Welcome to the world of telecommunications auditing. It’s unlike anything you’ve done so far, and more exciting and interesting than you can imagine.”

*The TL 9000 Guide for Auditors, p. 2*
1 Introduction

In today’s fast-paced world, there is a constant demand for better and more efficient ways of doing business. At the same time, a certain level of quality must be assured of the delivered goods and services. Motorola in Linköping realized they were in need of more efficient ways of developing software for their set top boxes. The solution was a transition to agile development with the framework Scrum. It emphasizes the importance of individuals, working software, customer collaboration, and responsiveness over processes, documentation, contracts, and elaborate plans.

However, the customers and the company expect the TL 9000 standard, the telecom industry extension of ISO 9000, to be followed in order to assure the quality throughout production. Some sort of conflict could potentially arise between these two approaches. The question is if there is a mutually beneficial way the approaches may be combined in the same organization.

1.1 The Case

Motorola was founded in 1928 in Chicago under the name Galvin Manufacturing Company. Radio power supplies were one of the first products, followed by two-way radios, pagers, and cellular phones. The name was later changed to Motorola and the company is today one of the leaders in mobile communication technology, with approximately 66,000 employees around the world. (Motorola, 2008a; Motorola, 2008b).

In 2006, Motorola acquired the Linköping based company Kreatel AB. Specializing in set top boxes and software platforms for IPTV systems, Kreatel provided Motorola with software development expertise and a position as a leader in the growing market. The customers are mainly large telecommunication providers in North America, Europe, and Asia. (Thulin, 2007)

Motorola have previously been using a project model based on a traditional waterfall model. However, in mid 2008, the Research and Development department in Linköping adopted the Scrum framework in their software development process. This was due to the perceived flexibility and adaptivity Scrum offered compared to the previously used model.

TL 9000 is a global quality assurance standard for telecommunication companies, which is vital to the customers and it is also a demand from the company, Motorola. Motorola in Linköping is today certified by TL 9000. During the certification process, called an audit, two major steppingstones were used. One of the steppingstones was the agile development framework Scrum, and the other is a framework of procedures called a product-life-cycle-model. Together these two ensures that the development process is compliant to the TL 9000 requirements. The product-life-cycle-model is common for all business units within Motorola and is based on a waterfall model, while Scrum is used locally in Linköping. Some required activities from the model are not a natural part of the agile development process and could therefore be seen as a waste of effort. This prompted management at Motorola to question if the model is really needed or if the TL 9000 standard could be met by Scrum alone.

---

1 IPTV, or broadband television, is digital television distributed through an IP network instead of satellite, cable, or terrestrial.
The topic of this thesis was suggested by Motorola in Linköping, based on their situation and interest in working efficiently. At the same time, the topic was independently suggested by two other major companies also located in Mjärdevi, currently facing similar problems. This indicates a large interest in the transition to agile development, like Scrum, while still adhering to formal quality assurance standards, like TL 9000.

1.2 Aim
The aim of the thesis is to investigate if it is possible to achieve a quality certificate from TL 9000, the telecom industry extension of ISO 9000, when using the agile development framework Scrum. By interviewing employees and observing how Scrum is adopted at Motorola, the use of Scrum will be inspected. At the same time quality management and agile development must be examined through literature studies. The adoption of Scrum at Motorola will be compared to the requirements of TL 9000, and the non-conformances can be avoided through suggestions for improvements, that are founded in quality management theories. In order to facilitate the investigation, the aim is divided into smaller parts through questions in a problem definition.

1.3 Problem definition
The master thesis aims to answer the following questions:

- How could Motorola meet the TL 9000 requirements by using Scrum?
  - What is quality management?
  - What are the requirements of TL 9000?
  - How is Scrum explained in existing literature?
  - How does Motorola use Scrum?
- What are the requirements of TL 9000 that Scrum contradicts or does not deal with?
- Is it possible to make changes or additions to Scrum so that it satisfies these requirements?

1.4 Limitations of Scope
This master thesis is only treating the problem description within the theoretical frame of reference: Quality Theory and Agile Development. The empirical studies will solely be performed at the Research and development department at Motorola Linköping. Hence, the empirical section of the master thesis will strictly focus on using agile development of software for set top boxes. Agile development is performed using Scrum, which will be compared against the quality assurance standard TL 9000.

The secrecy of material belonging to Motorola also limits the thesis. Since a master thesis is published and shown to the public, classified information must not be used anywhere in the thesis.

---

2 Mjärdevi Science Park is the centre for the research and development companies in Linköping, Sweden.
2 Method
In this section, different research strategies and orientations are examined and discussed. The different methods will be described and carefully compared to each other before an approach is chosen.

2.1 Methodical orientation
One of the most fundamental questions to ask when writing a thesis is what scientific approach to use. According to Bryman (2004), there are several ways to relate to knowledge and research.

Positivism is usually thought of as an attitude towards the theory of knowledge where only a phenomenon observed and verified through the senses is considered knowledge. It is usually applied to scientific research in natural science and aims to test a hypothesis based on previous knowledge that can be proved or disproved objectively from the surrounding research environment. The test should give the same result every time it is performed, even if the scientist is replaced, and the result could therefore be extended to a universal explanation. (Bryman, 2004)

Hermeneutic theory stands in the opposite of Positivism, since it claims that the findings needs to, and will be, interpreted before any conclusions can be drawn. The Hermeneutic theory is used in several different scientific disciplines, but foremost in human-, culture-, and social science. A Hermeneutist argues that human reality can be interpreted through the language. The Hermeneutic approach of gaining knowledge is through observation and interpretation of human actions. (Patel & Davidson, 2003)

Since the scope of this master thesis is closely tied to personal opinions and human interaction, a hermeneutic orientation is most suited. This master thesis includes interviews, observations, and literature studies. All of these are affected by a persons understanding and personal opinions about the subject. It is impossible for a person to read literature and not be affected by personal opinions and previously knowledge in relation to the subject, since reading demands a certain amount of interpretation.

2.2 Qualitative or Quantitative research strategy
It is common to categorize a study as qualitative or quantitative. A qualitative study is most often used to achieve a greater comprehension on a specific subject, a specific issue, or a specific situation. The qualitative study can for example be interviews and observations and it is used to answer questions that are more complex and when a deeper understanding for a specific subject is desired. Quantitative research most often demands a greater amount of interpretation. Quantitative research strategies are however mostly used when the studied information is numerically measurable. Examples of quantitative research methods are questionnaires and surveys. (Björklund & Paulsson, 2003)

To determine whether qualitative research or quantitative research should be used, the problem definition can be used. If the problem is defined as: Where? How? or What are the differences? a quantitative research method is appropriate to use. If the problem is about interpreting and understanding for example peoples opinions. Typical questions are: What is this? or What are the underlying patterns? The two
methods are however not contradicting each other completely, it is possible to perform research founded somewhere in-between the two methods. (Patel & Davidson, 2003)

2.3 Triangulation

When creating the interview questions, books about interview techniques is used in order to make the questions as clear and obvious and as understandable as possible (see chapter 2.5.2 Interviews). By using several different and independent sources and/or methods to verify the same phenomenon, the validity is improved. This technique is known as triangulation and results in a more complete answer. The results from different methods or sources may either coincide or differ. If the result differs, additional investigations are needed. If the results coincide it is most likely a true picture of the situation, and it is these results that are the most interesting in the study. (Patel & Davidson, 2003)

2.4 Case study

Yin (2007) describes a case study as:

"An empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.” (Yin, 2007, p. 31)

The case study is a method useful when the context of a phenomenon is interesting to study in itself, as opposed to for example an experiment where the phenomenon and the context should be separated. Case study as a research strategy embodies all aspects of research, including test design, data gathering techniques, and approaches to data analysis. (Yin, 2007)

A case study is usually restricted to a specific location, such as a community or organization, of a special interest, which is studied intensively. Most often, a qualitative approach is used; however, quantitative methods could possibly be used as well. (Bryman, 2004)

When designing a case study it is possible to take two different approaches: single- or multiple-case study. Multi-case designs combine different case studies into a single analysis, where single-case design concentrates on a single case. Multi-case designs are usually preferred because of the added analytical benefit the comparison of two or more cases can give. However more resources, especially time, and knowledge, how to choose and analyze the different cases, are generally required. (Yin, 2007)

There are also different types of single-case studies:

- The **critical case** is described as a unique case, against which a previously suggested theory is tested in order to question or develop the theory.
- The **unique case** is either something that is extreme compared to the environment, or a rare occurrence, and interesting to document and study.
- The **typical case** is on contrary something that is believed to represent the ordinary and common. This is used to describe one case and generalize the finding to other similar cases.
• The *revealing case* is a case where the researcher has an opportunity to study an environment that might well be common, but has previously been unavailable for one or another reason. Such a case study could give new insights.

• The *longitudinal case* is when one case is studied at two or more times. The underlying theory should preferably specify how factors change over time and the time span for the study should be chosen in accordance with the theory. (Yin, 2007)

When preparing a case study, something called a case study protocol is written. It contains the procedures and general rules to be followed by the researcher, together with more specific questions that is the overall aim of the research. These questions are not used during interviews but are a reminder to the researcher about the important topics in the study. Through the protocol the reliability of the study is increased since it documents the way the study was performed, enabling other researchers to replicate it later. It is also especially useful in multi-case designs where it ensures that the same questions are asked in the different cases. (Yin, 2007)

Six different types of sources can be used in a case study: documents, archival records, interviews, direct observation, participant-observation, and physical artifacts. They are all useful in different situations depending on the context and aim of the study. More than one source of information should generally be used to give a more balanced view of the subject. (Yin, 2007)

Documents and archival records are persistent and contain a lot of information, but might also be hard to find or gain access to and must be interpreted by the researcher. Interviews might give answers to specific questions related to the study, but the answers might be biased because of the interaction between interviewer and interviewee. Direct observation and participant-observation describes real-life events in its context, but takes time and events might be altered because of the presence of observers. Physical artifacts describe cultural or technical aspects, but might not be accessible or relevant enough. (Yin, 2007)

The question about anonymity is common in case studies. Should the name and location for the entire study be revealed, and should individuals participating in the case study be identified. The most desirable option is to disclose all available information. This enables the reader to use any previous knowledge about the case, and to verify and check the details of the study. However, sometimes anonymity is necessary. Studies on controversial topics and cases where the final report might affect the subsequent actions of the participants are examples where anonymity is required. The possibility of individual anonymity should always be considered first, still leaving the case to be identified. It should be noted that even when individuals are anonymized, some people with certain knowledge will still be able to identify them. (Yin, 2007)

2.4.1 Reliability and Validity

Four criteria are commonly applied to all research methods in order to test their quality. These are: construct validity, internal validity, external validity, and reliability. The difference between reliability and validity will be explained and illustrated in Figure 1. (Yin, 2007)
Reliability is used in testing and measurements, and is a measure of the authenticity of the test or measurement. For example, it is not especially reliable to measure long distances by pacing with your feet. Thus, pacing is a measurement with low reliability. In this case, a measuring-tape would have given a result with higher reliability. (Ejvegård, 1996)

Reliability ensures that another researcher should be able to do the same research and end up with the same result. This requires the methodology to be well documented and the research conducted in a way that could be repeated. (Yin, 2007)

Validity states whether a measurement really measures what it is intended to (Bryman, 2004). For example, if an intelligence test measures a person’s memory capacity, it measures an important aspect of intelligence but not all parts. Thus, this intelligence test has low validity since it does not present a complete measure of a person’s intelligence. (Eriksson & Wiedersheim-Paul, 2006)

![Figure 1. Reliability and validity using arrows on a dartboard](Adopted from Björklund and Paulsson (2003) p. 60)


Construct validity tests if the definition of the subject of the research is clearly defined. Otherwise, it is hard to know if the result actually presents any objective evidence or if it is just the subjective opinion of the researcher that has been given a scientific look. Why a certain aspect has been studied and why the selected measures are valid are two important questions the researcher has to answer in the report. Using multiple sources of evidence results in a chain of evidence that can be followed and understood by the reader. Key informants review the draft to ensure that the results are in line with their description. (Yin, 2007)

If the research infers that an event is based on an earlier occurrence, internal validity has to be considered. Could there be another theory that leads to the same result or a rival explanation. (Yin, 2007)

External validity is related to the generalization of the result. Unlike for example a survey, the result from a case study is not statistically generalizable, with the particular case seen as one sample, to all populations. Instead, the result should be expanded and generalized into a theory, which can later be proved or disproved by replicated cases in different environments. This is similar to the way experiments are used in science. (Yin, 2007)
2.4.2 Case Study Protocol
This Case Study Protocol is outlining the approach to the Case Study we will perform. The Protocol is used to guide and control our efforts to be in line with the assumptions and expectations set at the start of the study. The purpose, scope and problem definition of the Case Study is the same as the Master thesis as a whole.

Data Collection Procedures
The Case Study will be performed at Motorola Linköping during 20 weeks from September 2008 to February 2009. Information will be gathered through literature studies, observations, and interviews. The interviewees will be chosen depending on their role in the company. Being well prepared before performing the interviews is important as well as having well thought-out and relevant questions. The observations can be used to form appropriate and relevant interview questions and function as preparations for the interviews.

Case Study Questions
These are the Case Study questions and our ideas where we might find relevant information. These questions are a help for the interviewers and should not be used as interview questions. The Case Study questions are brought from the problem definition (see 1.3 Problem definition).

- **How could Motorola meet the TL 9000 requirements by using Scrum?**
  This can be determined by making conclusions from the answers in the four questions below.
  
  - **What is quality management?**
    There is a big amount of information about quality management in existing literature. They should be more than enough to answer this question.
  
  - **What are the requirements of TL 9000?**
    By reading the TL 9000 Requirements Handbook, we gain understanding of the formal requirements in detail. This could be contrasted to interviewing someone with experience in interpreting the requirements and applying them in real life. To interpret and understand the requirements correctly, more experienced people at Motorola can be consulted for a clarification.
  
  - **How is Scrum explained in existing literature?**
    Scrum is described in detail in the various books about the process. Notably the ones written by Ken Schwaber, one of the driving forces behind Scrum, should form a foundation for the rest of the work.
  
  - **How does Motorola use Scrum?**
    Information about this question could be obtained by interviewing and observing different employees in the company and ask them about their work. The observations will form the foundation on which the interview questions will be based and the interviews will be the documented and most relevant source to answer this question. If any relevant written material is available, this could also be a good source.
• **What are the requirements of TL 9000 that Scrum contradicts or does not deal with?**
These are the requirements that are contradicting between TL 9000 and Scrum, or is not treated at all by Scrum. We will be able to answer this question by comparing the TL 9000 requirements to Scrum, and thereafter determining whether the requirements are fulfilled or not. These are also found through examination of the TL 9000 requirements and the Scrum implementation at Motorola.

• **Is it possible to make changes or additions to Scrum so that it satisfies these requirements?**
By using the previously gained knowledge about the Scrum theory, in conjunction with the TL 9000 theory we will be able to analyze and discuss the question. This could also be compared to the current implementation of Scrum at Motorola.

**Outline of Case Study Report**
The result of the Case Study will be analyzed, discussed, and presented as conclusions in this Master Thesis report.

### 2.5 Data gathering techniques
This thesis will be based on a literature study and interviews. To give a better understanding of the subject observations will also be made.

#### 2.5.1 Literature study
According to Björklund and Paulsson (2003) all forms of written material is considered literature, including books and articles. A study of the available literature can give a good overview of a subject in a relatively quick and inexpensive way. It can therefore be used to provide a theoretical frame of reference, outlining the existing knowledge in the field. However, different literature, even within the same subject, is not necessarily written with the same purpose and the research methods used are not always clearly defined. Therefore, it is important to have a critical attitude towards all literature. (Björklund & Paulsson, 2003)

Assumptions must be identified and questioned, and results from different sources must be compared. Different researchers will also have different opinions and understanding of the subject in question, something that is reflected in the selection of literature used for the research. (Bell, 2006)

Even though books are the most common source of written text, the most recent research is usually found in articles. Some minor or developing subjects are not even covered in books at all. Relevant literature can be found by searching library catalogues and online databases with carefully selected keywords. Finding relevant keywords is usually the hardest part and requires some understanding of the field of research. Another approach is to use reference chaining. One source is identified, usually a recent article about a subject as close as possible and the literature it references is then considered. If a reference is found to be relevant to the research aim the same procedure is repeated on that source. This ultimately generates a chain where each author has been recommended by another. (Rienecker & Jørgensen, 2002)
2.5.2 Interviews

Most people have some kind of experience of interviews, for example from listening to radio and television or participating in a job interview. It is important to be aware that what is being said in an interview will always be subject to interpretation, and it is possible to formulate questions in such a way that a certain answer is more likely to be given. (Lantz, 2007)

When using an interview for research purposes a critical view of the way it was performed must be taken. There are usually three requirements on the method that shall be fulfilled: it must give reliable results, the results must be valid, and enough information should be given to enable a critical review of the interview process. (Lantz, 2007)

The result of a good interview is a true account of the source. The interviewer should try to ask relevant questions and capture the interviewee’s opinions and feelings at the time of the interview. It is however, according to the scientific approach, impossible to give an objective description of the interviewee’s account. This is not necessary; instead, a subjective description that reflects the source in the best possible way should be the aim. (Lantz, 2007)

According to Lantz (2007), it is important to have a well-defined research aim and a theoretical foundation before the interviews are carried out. There is otherwise a risk that important questions will be omitted and unnecessary data will be collected, mistakes that are easy to avoid with proper planning. The theoretical foundation also gives the interviewer an understanding of the subject and helps defining the questions. Since it is not possible to carry out an interview without being biased in some way, by for example deciding between asking and not asking a certain question, it is important to be able to know what questions might be relevant according to the theory. The theory can also help the researcher evade topics that are not relevant to the research. (Lantz, 2007)

The selection of interviewees is an important part of qualitative research, one aspect being the number of them. There is an upper limit on how many interviews that can be carried out due to the amount of time it takes to analyze them. At the same time, the number of interviews must not be too small if a balanced view of the subject is to be given. (Dalen, 2008)

The selection can be done in several ways but it is necessary to have a strategy and present it. In theoretical selection the maximum variance in the field is identified and studied, for example people with radically different opinions of a subject is interviewed. It gives a deeper understanding and increases the validity of the research, but requires the researcher to be well familiar with the subject of study. (Dalen, 2008)

If this is not the case, Dalen suggests criteria selection as a solution. Instead of identifying where the maximum variance could be found, some generally valid selection criterion is identified. Even though the research method is still qualitative interviews, the method used for selection borrows some techniques from the quantitative research. If for example the selected interviewees do not represent the
target group in general, additional interviews must be done in order to compensate for the bias. (Dalen, 2008)

Different target groups, for example both employees and upper level management, could be interviewed to give a better understanding of different point of views. Dalen stress that many researchers forgets to mention how the selection of interviewees was done and therefore makes it hard to determine the validity of the research. (Dalen, 2008)

An interview can be done as an open interview, a structured interview, or something in between. In an open interview, the questions typically explore how things are and what meaning they have. Different interviews will most likely end up in completely different results even if the same questions are used. The subjective opinion and context of the interviewee is reflected in an open interview. (Lantz, 2007)

A structured interview is more like a survey or questionnaire, asking to what degree a predefined statement is true. The topic and context is already defined through the questions, based on the researcher’s previous understanding of the subject and the theoretical foundation. (Lantz, 2007)

Semi-structured interviews are less formalized than a structured interview and more directed than an open interview. The interviewer often uses a guide, a list of specific topics that should be mentioned during the interview, but the interviewee is free to shape the answers in a personal way. If the discussion is brought on to an interesting topic, the interviewer can choose to explore that further with new questions. The guide should not be too specific since that might hold back alternative viewpoints and restrict the interviewee. (Bryman, 2004)

An interview might result in different answers from different persons, even if the same questions are asked, since the interpretation and the context differs. The answers might also be interpreted differently based on the researcher’s assumptions and previous knowledge. (Lantz, 2007)

If the questions are of a sensitive nature the interviewees is likely to wish their identity not to be revealed. If there is any risk that the answers will harm them, confidentiality is necessary. However, in qualitative research it might be hard to provide true confidentiality since the identities could possibly be revealed through context and descriptions. In, for example, a company context it might not be possible for an outsider to identify the anonymized persons but a person working in the company might identify the person from description. (Bryman, 2004)

2.6 Work process
In order to fulfill the aim of this thesis our research strategy will include a combination of literature studies and gathering of empirical data. The theoretical framework will have a strong foundation in quality assurance in general and formal quality assurance standards in particular. This will be compared to literature on organizational theory and project management, especially about agile development and particularly Scrum.
We will investigate if theories on how to combine agile development and formal quality assurance standards already exists. If not, a full comparison between the requirements of TL 9000 and Scrum must be done. This will form the basis for the development of our theory and how the two could be combined.

The theoretical version of Scrum will also be compared to Motorola’s adaptation, represented by empirical data gathered through interviews, questionnaires, and observations. By comparing the theoretical framework, how to make Scrum compatible with TL 9000, to the actual adaptation we hope to highlight the changes needed, if any, to make the theory work in practice. The intersection between theory and practice will also serve as ground for further analysis of the problems and opportunities that will arise. This however requires a certain amount of interpretation of the TL 9000 requirements. The requirements are not stated in plain text and only a brief description of the requirement will be presented.

Figure 2 illustrates the proposed solution path for this thesis. There are two major branches of underlying theory: Quality assurance management and agile development. These two branches will be investigated by using literature, interviews, and observations. Furthermore, the two branches will be compared. This comparison results in a description of fulfilled and unfulfilled requirement, which will be the base for suggestions of improvements.

![Diagram](image)

**Figure 2. Thesis solution path**

### 2.7 Methods used for this study

This study is based on a hermeneutic view, which is believed to be most appropriate in an environment with human interaction and different personal opinions. A qualitative approach is taken because of the complexity of the research questions and the purpose. Since the theoretical studies forms the foundation of this study a deductive approach I chosen. The study is performed in the form of a case study with a typical case used to illustrate the implementation of Scrum in an organization.
The theoretical studies in this master thesis will form the foundation and the base, which the empirical studies will be applied and compared to. For that reason, a deductive approach is chosen for this thesis.

A literature study is used to provide background information about quality, Scrum, and other relevant topics. Keyword searching is giving a broad overview that combined with reference chaining highlights the authorities on the subject.

A number of anonymous interviews are performed to gather information from employees in the company. The interviewees are chosen with criteria selection as representatives of the entire population of people involved in the software development. After the interview, all interviewees receive a summary of the interviewee’s notes together with the option to add or remove anything from their answers. This is done in order to increase the validity by making sure that the interpretation of the interview is as accurate as possible.

Observations are made throughout the study to give a basic understanding to select relevant questions before and during the interviews. They are also used to describe activities not mentioned in the interviews.

By using several sources within each method and comparing different methods, the validity of the result is increased.

2.7.1 Interviews
The aim of this part is to describe how Motorola currently use Scrum in their software development of set top boxes in Linköping. Interviews with employees were performed during a period of two weeks. The interviewees were selected based on a number of attributes (as described in 2.4.2 Case Study Protocol), and asked about their participation in an interview related to Scrum and TL 9000 for approximately thirty minutes. A few of the first selected interviewees declined, and another person with similar attributes were chosen instead. Because of this, in one case two members of the same team had to be chosen. In total twelve persons were interviewed: eight representatives from the seven Scrum Teams involved in software development, two line managers and two project managers.

Each of the interviews started with a brief background to the study and an explanation of how the interview was going to be carried out. The anonymity of the interviewee was also guaranteed. The interviews were performed in a semi-structured form, with a list of questions forming the basis for the interview and additional questions added to highlight interesting topics. Notes were taken continuously by both interviewers regarding the answers given by the interviewee.

After the interviews, the notes were combined and summarized for each interview. In order to increase the validity, all interviewees were given the opportunity to read a summary of their own interview to help minimize misunderstandings and give the interviewees a chance to adjust their statements. If requested the notes were corrected and the updated versions used for the summary instead. However, only a few of the interviewees actually had any comments on the notes other than an agreement.
In addition to the interviews, the result is based on direct observations performed on several occasions during a period of a few months. The observations have focused on the different practices in Scrum: Sprint Planning Meeting, Daily Scrum, Scrum of Scrums, and Sprint Demonstration. Different behavioral aspects, like how different teams approach a new task or conduct their Daily Scrum meeting, have been noted together with environmental aspects, such as the layout and design of the different workspaces.

The interviewees all have different backgrounds and work experience, ranging from ten years full time employment in the company to working a few months as a consultant. Nearly everyone has some kind of computer related university degree, most commonly from Linköping University.
3 Quality
This chapter explains the foundations in producing quality products and some important and widely used quality tools are introduced.

3.1 Quality Theory
The word quality originates from the Latin word “qualitas” which was used already in the classical era to describe the condition of goods (Sandholm, 2000). During the past fifty years, quality has been a dominant theme in management thinking. The modern quality thinking started in America but the Japanese companies early adopted and developed the theories. While quality was more or less rejected in the West, it grew in the East. In recent times, organizations throughout the world have embraced and further developed the Japanese’s work. (Beckford, 2002)

It is common to talk about how different types of products have various qualities. High quality is desirable in the pursuit of organizational effectiveness and success. There are numerous definitions of quality and there are many writers that have underlined the importance of quality. These are definitions given by some of the gurus on the subject:

- Kaoru Ishikawa: “quality of product, service, management, the company itself and the human being”
- Joseph M. Juran: “fitness for use or purpose”
- John S. Oakland: “quality is meeting the customer’s requirements”

Ishikawa focuses on the whole chain of production while Juran and Oakland more focuses on the complete product and what the user experience. These definitions of quality are widely recognized and used nowadays. (Beckford, 2002)

There are different quality guidelines, such as ISO 9000 and Six Sigma, which helps improve quality in different ways. They also contain definitions of quality:

- ISO 9000: “the degree to which a set of inherent characteristics fulfils requirements” (Hoyle, 2003). A requirement is then defined as “need or expectation that is stated, generally implied, or obligatory” (Bergman & Klefsjö, 2007).
- Six Sigma: “Quality is a state in which value entitlement is realized for the customer and provider in every aspect of the business relationship” (Bergman & Klefsjö, 2007)

Since a part of the scope for this Master Thesis is to investigate how Motorola could meet the TL 9000 requirements, the TL 9000 definition of quality is most suitable. TL 9000 is an addition to ISO 9000, and TL 9000 thereby inherits the quality definition from ISO 9000. Thus, the definition of quality for this Master Thesis will be “the degree to which a set of inherent characteristics fulfils requirements”.

3.1.1 Quality Management Systems
A Quality Management System (QMS) is a formal record of how an organization handles the quality of its products and services. This enables the organization to show
itself, its customers, and maybe most importantly: an independent accreditation organization, evidence that the quality is handled in a structured and effective way. (Beckford, 2002)

Quality assurance procedures were first introduced in industries where relatively small errors could have disastrous consequences, such as defense, space, aviation, and nuclear technology. The idea of QMS was an important improvement that was first adopted in the US defense forces during the 1950s. Nowadays it is common in all types of industries. (Karlsson & Söderstedt, 1997).

One of the underlying reasons for a QMS was to get rid of the work intensive practice of customers sending their own inspectors to check the quality at the supplier. Not only did it force the customer to hire staff for the inspections; at the customer’s side, the additional workload and lockup of key personnel during the inspections could potentially create more problems than they solved. By using a QMS, the supplier could instead give evidence that they are meeting a certain quality standard. Through regular assessment by an accreditation organization, which if the result is positive will issue a certificate, it is easily proven that the quality system complies with the requirements. (Sandholm, 2000)

Today there are several reasons for a company to have a certified QMS. Internally it provides a formalized description of how quality activities should be performed to produce consistent output. Many customers have a policy of only engaging suppliers with a QMS, which gives certified organizations a competitive advantage in the market. In some industries, there are also requirements from authorities that a formalized and well-documented quality system should be in place. (Sandholm, 2000)

Three levels of different quality control are usually identified. Complex products demand control and inspection of all activities and functions. Less complex products demand control and inspection of the production while simple products only require inspection and testing. (Sandholm, 2000)

3.1.2 Cornerstones in Total Quality Management
Today, many companies and organizations see quality as an integrated part of their business. Based on this Bergman and Klefsjö (2007) describes Total Quality Management, the organization wide management of quality. Besides committed and dedicated leadership, five cornerstones are described: Putting the customer in the centre, Base decisions on facts, Work with processes, Continuously work with improvements and Create conditions for involvement. (Bergman & Klefsjö, 2007)

Putting the customer in the centre
The quality must be evaluated by the customers and it must be compared to their needs and expectations. In this aspect quality is a relative concept, which is formed by the competition in the market. Putting the customer in the centre means finding what the customer wants and during development systematically and continuously meet and exceed their needs and expectations. (Bergman & Klefsjö, 2007)

Base decisions on facts
In order to not make decisions based on chance this cornerstone is of great importance. To do so, knowledge and information is needed, and facts must exist to
be analyzed. These must be gathered, structured, and analyzed, and there are many different tools, such as diagrams and graphs, to facilitate the structure and analysis. (Bergman & Klefşjö, 2007)

**Work with processes**
A process is a network of interconnected activities that is repeated over time. The purpose of the process is to satisfy customers by transforming resources into result and at the same time use as little resources as possible. To make the process as optimal as possible, it is important that all the suppliers to the process are identified. The suppliers are given clear instructions of what is expected to be delivered, to minimize the waste of resources, and to satisfy the customer needs. It is common to distinguish three separate types of processes: Head process, Support process and Management process. The Support process and Management process supports the Head process, which is supposed to fulfill the needs of the external customers. (Bergman & Klefşjö, 2007)

**Continuously work with improvement**
This cornerstone is based on the statement “What fails to improve, will soon fail to be good”. The requirements on quality increases steadily, therefore continuous quality improvements are essential. This is also motivated from an economical viewpoint, since the measured expenses caused by lack in quality leads to large monetary losses. It is not unusual that lack of quality costs about 10-30% of the total volume of sales. One way to adopt this is to use the method Six Sigma, explained in Fel! Hittar inte referenskälla.. (Bergman & Klefşjö, 2007)

Create conditions for involvement
To successfully perform quality work it is important to facilitate participation of all co-workers. By promoting communication, delegation, and education, co-workers can actively affect decisions and participating in the improvement process. When a person is given freedom and responsibility, he or she works the best. If a person is given the conditions of doing a good job and to take pride in the work, he or she will be more engaged in it. This will contribute to the improvement of quality in the processes and the product. This also implies that the individual must take its responsibility and it demands high performance in for example communication and trust. Bergman and Klefşjö describe the effects of involving co-workers versus not involving co-workers in the good and the bad circle (see Figure 3).

Figure 3. Bad and good circle of motivation affection of the result
(Adopted from Bergman & Klefşjö, 2007, p. 49)
It is important to turn the bad circles into good; not only because of the improved result, but also because of it is also important for the co-workers to have a satisfying work to do as this improves quality. (Bergman & Klefsjö, 2007)

**Dedicated Leadership**

It is maybe the most vital part of the Total Quality Management to have a strong and committed leadership. It is proven that the leadership affects the engagement and participation of the co-workers; therefore, the leader works as a role model. A good leadership makes the co-workers feel proud of their work and for the result of the organization. (Bergman & Klefsjö, 2007)

### 3.2 Quality Tools

There are many different tools for achieving quality, ISO 9000 and TL 9000 are two of them. This section introduces these two tools and a clarification of quality in software development.

#### 3.2.1 The ISO 9000 family

In 1987, the International Organization for Standardization, ISO, defined three standards intended to be used in the quality assurance process between customers and suppliers. Named ISO 9001, 9002 and 9003 they reflected the three levels mentioned above. In reality, only ISO 9001 and 9002 was used since ISO 9003 did not provide enough control for real world applications. (Sandholm, 2000)

In the year 2000, the latest major revision of the ISO quality standard was made. ISO 9001, 9002, and 9003 was consolidated into a single standard, ISO 9001:2000. The family of quality standards is still known as ISO 9000 though. This new version focuses more on the processes while keeping the original requirements on the products or services. (Beckford, 2002)

Several industries have developed their own standards, such as QS 9000 for the automobile industry and TL 9000 for the telecom industry, after finding that the ISO requirements were not sufficient for their application. (Sandholm, 2000)

ISO 9000 is based on eight principles that should direct the organization and form the base for the QMS, according to the standard. **Customer focus** requires the organization to take care of their customers by meeting their needs and aiming to exceed their expectations. **Leadership** gives the organization a common direction and creates an environment where employees are able to reach the goal. Through **Involvement of people**, the organization will have committed employees working together and using their abilities in the most efficient way. A **process approach** to activities and resources, and a **system approach to management** of these processes are the most efficient way of managing the organization. **Continual improvement** develops the adaptability of the organization and should be a permanent objective. A **factual approach to decision making** leads to informed and effective decisions. Finally **mutually beneficial supplier relationships** acknowledge that an organization and their suppliers depend on each other and a good relationship increases the ability to create value for both parties. (Bergman & Klefsjö, 2007)
To adopt ISO 9001:2000 the organization is required to identify the processes needed, determine their order and interaction, decide the criteria and methods needed to run them in an effective way and make sure the resources and information is available. The processes should also be monitored and analyzed, and corrective actions should be taken if needed. (Bergman & Klefsjö, 2007)

Some kind of documentation is usually needed to let the organization, and the auditor, know what should be done and trace what really have been done. A *quality plan* contains information about how the quality management system should be used together with a specific product, project, or contract. A *procedure* defines how a specific activity should be performed. (Bergman & Klefsjö, 2007)

However, even if documentation is necessary the extent should be adapted to different factors, like the size and complexity and the actual needs of the organization. According to the ISO standard, producing documentation should not be a purpose of its own but an activity that adds value. (Bergman & Klefsjö, 2007)

3.2.2 Quality in Software development

ISO 9000 was originally developed with the manufacturing industry in mind, and consequently focuses more on manufacturing than design. When constructing tangible products the design phase is usually a small part of the process compared to the actual manufacturing. In software development, the opposite is true: design is what requires time, planning, and knowledge, while production usually means simple automated copying. (Oskarsson & Glass, 1995)

According to Tian (2005), software systems usually have two quality expectations: that they do the right things and that they do the things right. Simple as it may sound; “right” is not always well defined and usually varies between the user, the producer, and the customer. Even when defined, quality defects will be present in the software due to mistakes and false assumptions. Quality assurance in software development therefore has an important part to play by preventing or removing errors. (Tian, 2005)

3.2.3 Telecommunication Quality Standard - TL 9000

TL 9000 is the Telecommunication Quality Standard designed to provide a quality management system to the telecom industry. For example, the standard is used to improve customer relationships and reduce costs. Just like many other quality assurance standards, TL 9000 is based on ISO 9001 with all its requirements intact. On top of that, telecom related additions, *adders*, are placed to cater for the domain specific needs. (Kempf, 2001)

The standard was originally designed by the QuEST Forum in 1998 and published in two handbooks: “TL 9000 Quality Management System Requirements Handbook” and “TL 9000 Quality Management System Measurements Handbook”. These are now available in release 4.0, and will be refereed to as “Requirements Handbook” (QuEST Forum, 2006b) and “Measurements Handbook” (QuEST Forum, 2006a). (Kempf, 2001)

---

3 QuEST Forum is acronym for Quality Excellence for Suppliers of Telecommunications Leadership Forum.
The Requirements Handbook describes the requirements for suppliers of telecommunications products, including hardware, software, and services. The Measurements Handbook defines performance measurements of progress and evaluating results of a quality management system implementation. (Kempf, 2001)

According to the Requirements Handbook (QuEST Forum, 2006b) TL 9000 is structured in layers. Figure 4 illustrates how the requirements from ISO 9001:2000 together with the additions from Measurements Handbook and Requirements Handbook forms the standard. All implementations use the common requirements and measurements that apply to both hardware, software and service providers. On top of that, specific requirements and measurements are added depending on the type of product or service being produced. (Kempf, 2001)

![Figure 4. The TL 9000 Layer model.](Adopted from Requirements Handbook (2006b) p. 2-1)

The Requirements Handbook (QuEST Forum 2006b) differentiates between mandatory requirements, *shall*, and preferred approaches, *should*. If another approach is used the organization must show that it still meets the intent of TL 9000. In the Requirements Handbook, all ISO 9001:2000 requirements are in italics with additional TL 9000 specific requirements below. (QuEST Forum 2006b)

### 3.2.4 TL 9000 Requirements

The requirements in the TL 9000 Requirements Handbook (QuEST Forum 2006b) are divided into five sections: Quality management system, Management responsibility, Resource management, Product realization, and Measurement, analysis, and improvement.

The rest of this section of the report refers to the Requirements Handbook. Only additions related to the common or software requirements (marked with C or S), not to hardware (H) or services (V), are included in this overview.

The first part, *Section 4 - Quality management system*, focuses on the general requirements of the QMS and the requirements on documentation, such as what the quality manual shall include and how documents and records shall be controlled. However, this section only contains one TL 9000 specific addition: [4.2.3.C.1] that deals with the control of customer-supplied documents and data.
In Section 5 - Management responsibility things such as the commitment of management, and how planning and reviews shall be performed, are described. This section contains a number of TL 9000 specific additions.

Requirement [5.2.C.1] describes that top management shall be actively involved in the customer relations, while [5.2.C.2] deals with how the communication with different customers shall be handled.

Requirement [5.4.1.C.1] is the largest extensions to ISO 9001, since it refers to and includes all the requirements from the Measurements Handbook into the standard.

Requirements [5.4.2.C.1], [5.4.2.C.2], and [5.4.2.C.3] deals with the QMS planning and customer and supplier input.

Requirement [5.5.3.C.1] states that the organization should inform the employees of its performance concerning quality.

Training and the handling of resources is described in Section 6 – Resource management. [6.2.2.C.1] requires the organization to ensure that quality standards are met on internal courses.

Requirement [6.2.2.C.2] states that employees influencing the quality of a product should have proper training, [6.2.2.C.3] that such training should be available, and that knowledge about ESD\(^4\) [6.2.2.C.4], advanced quality [6.2.2.C.5], and handling of hazardous conditions [6.2.2.C.6] should be included if appropriate.

Requirement [6.3.C.1] requires security for infrastructure critical to the organization, while [6.4.C.1] requires all relevant areas to be clean, safe, and organized.

Section 7 – Product realization describes different quality aspects of how the product is made. According to requirement [7.1.C.1] all products need to have an associated life cycle model, [7.1.C.2] a plan for disaster recovery to be in place, and [7.1.C.3] that documented procedures at the end of life for a product is maintained.

Internally developed software that is used in the process should be subject to appropriate quality methods, according to [7.1.C.4], and a plan for configuration management should be in place and used [7.1.HS.1].

Requirement [7.2.2.C.1] requires all actions resulting from requirements reviews to be tracked and [7.2.2.C.2] that a contract review process should be present and meet certain criteria.

If a problem is reported, customers who may be affected should be notified according to [7.2.3.C.1]. In most cases, the problem should be assigned a severity level based on the impact on the customer, that is used for deciding in what timeframe the problems needs to be fixed, according to [7.2.3.C.2]. [7.2.3.C.3] requires a formal protocol explaining how the responsibility of issue resolution is moved higher up in the organization when needed. The customer should get timely and systematic feedback about their problem reports according to [7.2.3.C.4]. A documented procedure to

---

\(^4\) ESD is acronym for Electrostatic Discharge
identify and recall unfit products from the market should exist according to [7.2.3.HS.1]. Customers should also be able to receive design and development process measurements upon request according to [7.2.3.HS.2].

The section 7.3 deals with the design and development, a subject highly relevant to the topic of this thesis. It starts, in [7.3.1.C.1], with the requirement of a project plan, which have to include such things as the project organization, roles, and methods to be used in the project. All documented requirements should be traceable [7.3.1.C.2], and the testing should be planned in advance [7.3.1.C.3]. There should also be a migration plan, [7.3.1.HS.1], if there are plans to migrate the system from an old to a new operational environment.

In [7.3.1.HS.2] it is stated that appropriate measures should be selected and used during the design and development planning, and in [7.3.1.S.1] that an integration plan must exist. It is also necessary to have methods for estimating different project factors [7.3.1.S.2] and the needed computer resources [7.3.1.S.3]. Regression tests, if performed, should be described in detail in the test plan according to [7.3.1.S.4].

Customers and suppliers should be able to provide input to the development activities [7.3.2.C.1] and the design and development requirements should be documented, including quality, safety and testing requirements [7.3.2.C.2].

In [7.3.2.C.3] the allocation of product requirements to architecture must be documented, and the software component requirements of the system must be determined, analyzed, and documented according to [7.3.2.S.1].

The design and development activities should result in system architecture, source code and user documentation among other things [7.3.3.HS.1].

The customer and/or user documentation must be verified before delivery according to [7.3.5.C.1]. The product must also be stress tested [7.3.5.HS.1] and tested for abnormal conditions [7.3.5.HS.2]. A system test must also be performed in accordance with the test plan [7.3.5.S.1].

A documented change management process used to track and manage changes including planning, testing and review must be in place according to [7.3.7.C.1]. When changes occur, customers must be informed [7.3.7.C.2] and problems and fixes must be handled by the configuration management system [7.3.7.C.3].

When purchasing products there must be an established and maintained documented procedure to follow, [7.4.1.C.1].

Requirement [7.5.1.C.1] describes that employees with customer contact should be provided with appropriate tools, training, and resources. [7.5.1.C.2] demands that the interference with the customer’s normal operations should be minimized during delivery and installation of products. In case of an emergency, [7.5.1.HS.1] states that there must be services and resources to support the recovery of products in the field. A plan for installation [7.5.1.HS.2], patching procedures [7.5.1.S.1] and documentation [7.5.1.S.2], and replication [7.5.1.S.3] must also be available.

In [7.5.3.HS.1] it is required that each product should be uniquely identifiable.
To preserve the product, requirement [7.5.5.C.1] states that protection from ESD should be used where applicable, products should be appropriately labeled [7.5.5.HS.1], and software virus protection should be considered where necessary [7.5.5.S.1].

The last requirement in this section, [7.6.C.1], demands monitoring and measuring devices to be identified and not used if they are unsuitable.

The last part, *Section 8 – Measurement, analysis and improvement*, describes how things should be measured and improved in order to increase the quality of the products. [8.2.1.C.1] requires the organization to collect data from customers concerning their product satisfaction.

According to [8.2.3.C.1] process measurements shall be identified, documented, and monitored at appropriate points.

[8.2.4.S.1] requires that software testing should follow the test plan and documented procedures, keeping records of test results and problems.

A trend analysis must be done on problems found in nonconforming products, according to [8.4.C.1]. Field performance data should be included in the QMS to help identify reasons for product failures [8.4.HS.1].

Requirement [8.5.1.C.1] includes requirements of a program for continual improvement of customer satisfaction, quality and reliability, and other processes within the organization. Employees must be encouraged in participating in this process, in accordance with [8.5.1.C.2].

Finally, [8.5.2.S.1] requires the organization to establish and maintain procedures for corrective actions once a trouble is diagnosed as a problem.
4 Scrum
In this chapter, an overview of agile development is given and the agile development framework Scrum is presented in detail.

4.1 Agile development
Agile development can be described as a flexible way of working (Gustavsson, 2007). According to Shore and Warden (2008), agile development is not a process; it is a philosophy, a way of thinking about software development. This refers to the Agile Manifesto, written in 2001 by the Agile Alliance consisting of 17 people with large experience of agile development. After discussing common features among agile development, the participants in the meeting signed a Manifesto of values. This is known as the Manifesto for Agile Software Development, but is often referred to as the Agile Manifesto (see Figure 5). (Cockburn, 2007)

<table>
<thead>
<tr>
<th>Manifesto for Agile Software Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:</td>
</tr>
<tr>
<td>• Individuals and interactions over processes and tools</td>
</tr>
<tr>
<td>• Working software over comprehensive documentation</td>
</tr>
<tr>
<td>• Customer collaboration over contract negotiation</td>
</tr>
<tr>
<td>• Responding to change over following a plan</td>
</tr>
<tr>
<td>That is, while there is value in the items on the right, we value the items on the left more.</td>
</tr>
</tbody>
</table>

Figure 5. Manifesto for Agile Software Development from Agile Alliance (Agile Alliance, 2001a)

These values are the fundamentals of agile development. The Agile Alliance has also written the “Principles behind the Agile Manifesto” (see Appendix A). Together with the above stated values, these documents shape the base for use of agile development. (Cockburn, 2007)

Cockburn explains that the manifesto is written by practitioners that have tried and found agile development during their work with software development. They use their experience in the field, not to make rules and regulations to others, but to develop practices that help themselves as well as others do a better job. The items on the right hand are not unimportant, the document only emphasizes that the left hand items are more important if you are forced to choose. (Cockburn, 2007)

The first point underlines the importance of seeing the people in the team and the interactions between them. The second point deals with the importance of software that is actually running, as compared to software that works on paper. The third point is concerned with the involvement of the customer in the whole development process, and to avoid thinking in terms of “us” and “them”. The fourth point is about adjusting
to quick changes, not just blindly following a plan that was prepared at the start of the project. (Cockburn, 2007)

According to Gustavsson (2007), agile development should be used when a result is needed fast, if the business or technological environment is unstable, or when the project description is very complex. There are however times when agile development is not effective to use. Examples include when the requirements are described in detail, the requirements are not changeable, or the cost of changing already delivered parts is very large. (Gustavsson, 2007)

There are several kinds of agile development, the most well known are XP (eXtreme Programming) and Scrum, but other types, such as Lean Development, are used as well (Softhouse, 2006). The relation between Scrum, XP, and Lean Software Development are explained further in 4.3.

4.2 Introduction to Scrum

Ken Schwaber is a leader of the agile process revolution and one of the signatories to the Agile Manifesto in 2001 (Schwaber, 2007). Schwaber is together with Jeff Sutherland the founder of Scrum and he has developed it to what it is today. He has also written several books about Scrum. In the introduction to one of his books, he explains Scrum like this:

> Scrum is a management and control process that cuts through complexity to focus on building software that meets business needs..."Scrum deals primarily at the level of the team. It enables people to work together effectively, and by doing so, it enables them to produce complex, sophisticated products. (Schwaber & Beedle, 2002, p. 2)

In another book, by Schwaber, two different faces of Scrum are explained:

> “I offer you Scrum, a most perplexing and paradoxical process for managing complex projects. On one hand, Scrum is disarmingly simple. The process, its practices, its artifacts, and its rules are few, straightforward, and easy to learn.”..."On the other hand, Scrum’s simplicity can be deceptive. Scrum is not a prescriptive process; it doesn’t describe what to do in every circumstance.” (Schwaber, 2004, p. xvii)

Briefly, Scrum is a term that describes a product development process (Schwaber, 2002). Scrum was first described in 1986 in the article “The new new product development game” in Harvard Business Review by Hirotaka Takeuchi and Ikujiro Nonaka (1986). The name Scrum was taken from a term in the sport Rugby:

> “In today’s fast-paced, fiercely competitive world of commercial new product development, speed and flexibility are essential. Companies are increasingly realizing that the old, sequential approach to developing new products simply won’t get the job done. Instead, companies in Japan and the United States are using a holistic
method—as in rugby, the ball gets passed within the team as it moves as a unit up the field.” (Takeuchi & Nonaka, 1986, p. 137)

Scrum is based on short iterations of about 2-4 weeks, small project teams, and daily status meetings. The idea is to have a working product at the end of every iteration, possibly with only limited functionality. A release is made after each Sprint. This could possibly be delivered as a final product. The project team adjusts the delivered functionality according to the complexity and time available, and uses feedback from the customer to focus their work on valuable functionality. Not every detail and task needs to be known at the start of the project, instead Scrum helps adapting to constantly changing requirements. (Schwaber, 2002)

4.3 What is Scrum?

The terminology used to define Scrum is not completely evident. The experts on the subject do not give a unified definition of Scrum and use different terminology about similar concepts. The confusion is based in unawareness among some of the experts, but also in disagreement of what Scrum really is. To avoid confusion in this thesis, a consequent terminology and a clear definition of the terminology will be used.

In existing literature, many different terms have been used to define or describe Scrum. These definitions (American Heritage, 2003) are the most frequently used to describe or define Scrum:

- **Method:** A means or manner of procedure, especially a regular and systematic way of accomplishing something.
- **Methodology:** A body of practices, procedures, and rules used by those who work in a discipline or engage in an inquiry; a set of working methods.
- **Praxis:** Practical application or exercise of a branch of learning.
- **Framework:** A structure for supporting or enclosing something else, especially a skeletal support used as the basis for something being constructed.

Gustafsson (2007) mention Scrum as a method as do Juell-Skielse (2007). There are also several other articles, talking about Scrum as a method, but there are about just as many mentioning Scrum as a methodology, for example Larsson (2008). Cockburn (2007) takes it one step further and is not only mentioning Scrum as a methodology, but even arguing for it. He defines a methodology as “A series of related methods or techniques” (Cockburn, 2007, p. 149) and declares Scrum a methodology. A methodology is, according to the definition above, a set of working methods and a method claims to be a systematic way of accomplishing something. Since Scrum does not give instructions on how to perform something, it should not be described as a method. Scrum should not either be described as a methodology, because a methodology consists of methods.

Ivar Jacobson, one of the critics of the use of Scrum, declares Scrum as praxis, since it needs other praxis’ in order to handle a whole project. He says Scrum is a great praxis but it cannot be used alone. Instead, Scrum is suggested to be combined with other praxis’. (Jacobson, 2008)

The founder of Scrum, Ken Schwaber, mentions Scrum as a “management and control process” in his first book about agile development (Schwaber, 2002, p. 2).
More recently, he has explicitly defined Scrum, “Scrum is a tool, a framework” (Schwaber, 2008 p. 2). The definition of a framework suits well together with what Scrum is. Scrum can provide support for software development but it will not define how you shall do it, just like a framework. Scrum is a skeleton that needs to be built upon, or a framework that needs to be filled.

The definition of Scrum as a framework will be adopted in this Master thesis. This definition may not be fully correct in all situations, which the confusion among the experts indicates. It is believed to be the most appropriate definition in this context.
4.4 Scrum roles

According to Schwaber (2004), there are only three roles in Scrum: the Scrum Master, the Product Owner, and the Scrum Team. Working together, the different roles complement and balance each other. The Scrum Master makes sure that the rules and values of Scrum is present within the organization, the Product Owner represents the customer and communicates the business goals, while the Scrum Team is a self-organizing unit that takes care of the actual work. (Schwaber, 2004)

4.4.1 The Scrum Team

The Scrum Team is a small, cross-functional team that performs all development. At the start of each Sprint, the team commits to a number of items from the Product Backlog that the team thinks they will be able to complete. The team then has the full authority to do whatever it finds necessary to fulfill the commitment within the Sprint. If, for example, assumptions about the tasks are proven invalid during the Sprint the team might still be able to meet the goals by reducing functionality. (Schwaber & Beedle, 2002)

The fact that the Scrum Team is self-organizing is a key principle in Scrum. Even if it is tempting for a Scrum Master to resolve the team's internal problems, helping them figuring out how to do it themselves by removing unnecessary responsibility and problems is usually the best practice. (Schwaber & Beedle, 2002)

The best size for a Scrum Team is seven people, according to Schwaber and Beedle (2002). Larger teams increases complexity, decreases productivity and make the Scrum practices hard to use. Smaller teams limit the amount of interaction and reduce the possible productivity gains. People with the different skills needed to meet the Sprint goal should be in the team, including analysts, designers, quality control, and coding engineers. (Schwaber & Beedle, 2002)

4.4.2 The Scrum Master

Schwaber and Beedle (2002) describe the Scrum Master as a fundamental part of Scrum, a guardian of its rules, values, and practices against external forces. The Scrum Master should also be a driving force behind the Scrum implementation and teach relevant lessons to team members. He or she also works closely together with the customer and management to select a Product Owner, and with management to build the necessary Scrum Teams, since the right type of people need to have these roles. When this is done, the Scrum Master helps the Product Owner and the Scrum Teams to create a list of outstanding work, known as the Product Backlog. (Schwaber & Beedle, 2002)

The Scrum Master teaches the Product Owner how to use the Product Backlog and other parts of Scrum as effectively as possible. If someone does not fulfill his or her role, the Scrum Master is held accountable since he or she has not taught how to do the job well enough. (Schwaber, 2007)

Unlike a traditional project manager, the Scrum Master does not assign tasks to different team members and have no authority over the development teams. Instead, he or she focuses on making sure that all the details of Scrum are fully functional within the team and organization. The Scrum Master could be compared to a sheepdog that would do anything to protect its flock, or team, removing any
impediments and disturbances to make them highly productive. The success of the project is one of the responsibilities of the role. The probability of success increases by helping the team turn the Backlog into functionality, as well as helping the Product Owner create and maintain the Backlog. (Schwaber, 2004)

During everyday work, the Scrum Master helps the Scrum Team by removing impediments that hinders their progress as well as making decisions, leading the Daily Scrum and improving productivity. Decision-making should be done instantly if necessary, without waiting for all information to be available. This helps the team focus on the task and know what to do at all times, even if more information becomes available at a later point and decisions may be changed. (Schwaber & Beedle, 2002)

### 4.4.3 The Product Owner

In a Scrum project, the customer is represented by the Product Owner. The main responsibility is to define what currently constitutes the highest priority business value and make sure the Scrum Team is working on that during every Sprint. This is done through adding and prioritizing different items of work to be done in the Product Backlog. After each Sprint, the Product Owner uses the result and feedback to assess if the priorities were right. If circumstances or business opportunities have changed, the Product Backlog could be updated accordingly before the next Sprint, dropping, adding, or reprioritizing items. (Schwaber, 2004)

It is important that only a single well-respected individual is Product Owner and responsible for updating the Product Backlog. Otherwise, Schwaber and Beedle (2002) argues, there is a risk that different versions of the list will appear depending on the different opinions on what is important, leading to confusion and frustration. The Product Owner also works together with the Scrum Team to estimate how long each item would take to complete. (Schwaber & Beedle, 2002)

The Scrum Team is not managed by the Product Owner in a traditional sense. He or she suggests what needs to be done at the start and evaluates the progress at the end of each Sprint, focusing on the customer business values. (Schwaber, 2004)

### 4.5 Practices

Compared to traditional project management, Scrum introduces some new principles and practices that need further explanation.

#### 4.5.1 Sprint

In Scrum, all work is done in iterations known as Sprints. Schwaber (2004) suggest a Sprint to last for 30 calendar days, but it could possibly be shorter. Every Sprint begins with a Sprint Planning meeting. The Product Owner meets with the team, describes the top priority items from the Product Backlog and gives them the opportunity to pick as many as they think they will be able to complete during the Sprint. (Schwaber, 2004)

After a number of items have been chosen the team creates a Sprint Backlog, the plan during the Sprint, and set the Sprint Goal. Each day a Daily Scrum is held to communicate the progress and efforts made in the team. At the end of the Sprint a Sprint Review session is held where the functionality from the completed items are demonstrated to everyone that is interested. The Sprint ends with a Sprint
Retrospective where the performance and lessons learned from the recent Sprint is discussed. (Schwaber, 2004)

![Project trade-off triangle](image)

**Figure 7. Project trade-off triangle**  

According to Dalcher and Brodie, a project is defined and limited by its financial budget, schedule, and the scope of the project. Figure 7 shows the trade-off triangle, which shows that a change to one side of the triangle has to result in a change of the other two correspondingly. (Dalcher & Brodie, 2007)

Schwaber and Beedle (2002) further divide the scope into two variables. That is, according to them four variables constrain all development projects: available time, cost, delivered quality and delivered functionality. During a Sprint, the first three are fixed, since its duration and hence the cost of salaries is known together with the organizational quality standard. The only thing that may be changed is the functionality, as long as it meets the Sprint Goal. By increasing or decreasing the scope of the functionality, the team is able to complete their assigned items. Any unfinished functionality that is deemed necessary is re-entered onto the Product Backlog at the end of the Sprint. (Schwaber & Beedle, 2002)

It is possible to cancel a Sprint in some cases. The Sprint Goal may become obsolete, due to changes in the market, business decisions, or other reasons. The team may realize mid-way that it cannot achieve the Sprint Goal, or it feels it needs more direction from management before proceeding with the implementation of more functionality. If someone outside the team tries to change the goal or nature of the Sprint, the team also has the power to cancel the Sprint, and call for a new Sprint Planning meeting. At that point, the reason for the cancellation will be explained, making people careful not to make any such changes. (Schwaber & Beedle, 2002)

**4.5.2 Self-organization**

One of the core ideas in Scrum is that the team is self-organizing. No one outside of the team is allowed to tell the team what to do once it has committed to a number of items from the Production Backlog at the start of the Sprint. Instead, the team assumes the responsibility of planning and reporting its work independently. Any resources, such as external consultants or new equipment, that the team thinks would
increase their productivity, should be available within budgetary constraints. (Schwaber, 2002)

Self-organization is, according to Schwaber (2004), something that is easy to learn on the intellectual level, but hard to use in practice. It takes some experience with Scrum before a team will understand how to take the responsibility and authority traditionally reserved for the management. The Scrum Master should help the team figure out how to organize itself, while at the same time avoiding the risk of taking their self-management away through explicit decisions. (Schwaber, 2004)

4.5.3 Sprint Planning Meeting
As mentioned earlier a Sprint starts with a Sprint Planning meeting. It is time-boxed to eight hours since the whole idea is to get work; not to think of working. The Sprint Planning meeting consists of two parts. In the first part the Product Owner and the Scrum Team gets together to decide what will be done for the coming Sprint. The Product Owner presents the most prioritized items from the Product Backlog to the team and tells them what is most desired. The Scrum Team questions the Product Owner about content, intentions, purpose, and meaning with the items. This is for the Scrum Team to understand the range and functionality of the item. When the Scrum Team knows enough to estimate how much they can turn into functionality during the next Sprint, they select that number of top priority items from the Product Backlog. The Product Owner is then informed about their decision and that the team will do their best to complete the assigned items within the duration of the Sprint. (Schwaber, 2004)

During the second part, the team plans the Sprint and breaks down the items into tasks. When the second part has started, the Sprint is started as well, and the clock is ticking towards the end of the time-boxed Sprint. The team is now responsible to manage themselves through the Sprint and they start by creating a Sprint Backlog, plan the Sprint, and set the Sprint goal. It is important for the team to have a Sprint goal to aim at in order to complete the tasks. (Schwaber, 2004)

4.5.4 Daily Scrum
One of the most fundamental parts of Scrum is the daily status meeting, known as Daily Scrum. During approximately 15 minutes every working day, the Scrum Team meets in a fixed location. The only thing that should be mentioned in the meeting is the team member’s answer to three important questions: what have you been doing since the last Daily Scrum, what will you be doing until the next Daily Scrum, and is anything hindering you from doing your work? (Schwaber & Beedle, 2002)

The location and time for the Daily Scrum should be fixed so everyone will know when and where to go every day. All members of the team are required to attend, but if it is not possible, alternatives include telephone attendance or representation by another team member. The meeting should start on time and latecomers are usually required to, for example, donate a small amount to charity. (Schwaber, 2004)

The Daily Scrum gives the whole group an update on what is going on in the project, and an opportunity to ask for help, input and feedback on the work. If any

5 Time-boxing is used to limit the duration of an activity. If an event is time-boxed to one hour, the event ends after one hour regardless of the result.
impediments are reported, it is the Scrum Master’s most important responsibility to remove them as soon as possible. The Daily Scrum should not turn into a working session or a place for long discussions between a few people. If this happens, a separate meeting should be held right after the Daily Scrum to address the issue. (Schwaber & Beedle, 2002)

In the Daily Scrum, people are divided into ‘chickens’ and ‘pigs’ depending on the role. Inspired by a fable about the chicken and pig discussing their future ham-and-egg business⁶, pigs are committed and accountable for the result while chickens are involved without being accountable. During the Daily Scrum, the pigs are at the center of attention, usually standing or sitting in a circle, reporting what is going on. A limited number of chickens are allowed to attend the meeting, standing outside of the circle, as long as they do not talk, disturb, or try to provide advice during or directly after the Daily Scrum. (Schwaber, 2004)

4.5.5 Sprint Review
When a Sprint is finished, a Sprint Review meeting is held. At the Sprint Review meeting, the Scrum teams present their completed items in a demonstration for the Product Owner and other Stakeholders that wish to attend. In order to present the item it must be defined as done. The Product Owner reviews if the items have the intended functionality. The result of the Sprint is inspected concerning the project goals and adaptations are made to maximize the chance of reaching the goals. This meeting is an informal meeting and it is time-boxed to four hours together with the Sprint Retrospective. (Schwaber, 2007)

The Scrum Team also reports other things they have experienced during the Sprint. This way others can embrace these experiences. The Scrum Master is responsible for setting up and coordinating the meeting, deciding how and by whom the result will be presented. The Scrum Master also notifies all attendants a week before the Sprint Review. No one should prepare extensively for the demonstration, and therefore are PowerPoint and similar presentations forbidden. (Schwaber, 2002)

4.5.6 Sprint Retrospective
After the Sprint Review, before a new Sprint starts, a Sprint Retrospective meeting is held. This meeting is used to evaluate the finished Sprint and is, similar to the Sprint Review, time-boxed to three hours. The Scrum Master here encourages the team to revise and improve the work process for future Sprints. This hopefully results in a more effective and enjoyable work-process for the next Sprint.

4.5.7 Working environment
To make it easier for people to communicate and facilitate the self-organization that is one of the core practices in Scrum, open working environments is highly recommended. Rolling desks and walls covered with whiteboards helps creativity and let the team rearrange the environment depending on whom they are working together with currently. In the same way, the team should be able to set its own working hours, as long as they are appropriate to the rest of the organization. Since the employees normally cost a lot more than their equipment, the best possible tools

⁶ The full story is found in different books and articles about Scrum, for example Schwaber (2004) p. 7.
4.6 Artifacts

Scrum introduces a number of new artifacts: Product Backlog, Sprint Backlog, Release Backlog, and Burn-down.

4.6.1 Product Backlog

The Product Backlog is a to-do-list considering the product and each team should not have more than one Product Backlog. It is a result of the work done by the Product Owner and contains a number of items with different priorities. Items are requirements and desired actions that will lead to improvement of the product, for example, new features or bugs needed to be fixed. The Product Owner gathers all items in the Product Backlog and the items are the basis for modifications of the product. It is important that each item adds value to the product and the item must be possible to deliver or demonstrate at a Sprint Review. The priorities are set by the Product Owner and the priorities decide in which order the items must be done. The Product Owner bases the ordering of priorities on the customer’s needs and the demands from the market. The Product Owner is responsible for keeping the estimates up-to-date and as reliable as possible (Schwaber, 2002). (Softhouse, 2006)

Since it is always possible to do some improvements to the product, the Product Backlog never will be empty or complete. The Product Backlog evolves when the environment of the product changes. When the customer’s needs and requirements change, the Product Backlog must also be changed. The estimates of items in the Product Backlog are more precise the higher priority they have. (Schwaber, 2007)

Figure 8. Relationship to the Product Backlog

Figure 8 show how the relationship is between the Product Backlog, the Scrum Team, and the Product Owner. Each Scrum Team must have exactly one Product Backlog to pick items from. Each Product Backlog must also be managed by exactly one Product Owner.

4.6.2 Sprint Backlog

A Sprint Backlog is a Backlog just like the Product Backlog, but the Sprint Backlog is limited in time to a Sprint and consists of all the items that are supposed to be done during the Sprint. The items for the current Sprint are the items from the Product Backlog with the highest priority. At the first day of the Sprint, the Scrum Team meets the Product Owner to formulate the final deliverables of the items. The teams break items down into smaller tasks. Tasks should have enough detail so that it is estimated to take 4-16 hours to finish, if not, the item should be divided to fit the time interval. Once the Sprint Backlog is set, only the Scrum Team can change it. (Schwaber, 2007)
4.6.3 Release Backlog
A Release Backlog is limited by the time before a release. It is a subset of the Product Backlog and consists of selected items for a release. For example when a new version of a product is released the Release Backlog contains items to be done before the release. The Release Backlog shall be estimated in days. (Schwaber & Beedle, 2002)

4.6.4 Burn-down and Burn-down chart
For each Sprint, a Burn-down is set up. It consists of all items that is planed to do for the current Sprint. When an item is completed, it is removed from the Burn-down, and the Burn-down decreases. When the Burn-down is empty, all the items in the Backlog are completed.

![Figure 9. A typical Burn-down-chart.](image)

A Burn-down is illustrated in a Burn-down chart, a graph starting at the sum of all estimated hours in the Backlog and when time is passing and items are completed the graph is declining. The Burn-down illustrates the estimated hours remaining in the Product Backlog, Sprint Backlog, or Release Backlog at any point in time. This shows how long time the project will last and a typical Burn-down chart is illustrated in Figure 9. (Softhouse, 2006)

4.7 Using Scrum in general
The flow of the Scrum framework used in practice is indicated by Figure 10. Top priority items are taken from the Product Backlog to the Sprint Backlog, where they are being worked on during the entire Sprint. Each sprint is in turn divided into days, with Daily Sprint being held at the start of every one. After a number of Sprints, the customer accepts the release, and a Final Product is delivered.
Using Scrum in a real environment requires the circumstances to be favorable for agile development. As with all other approaches to project management, there are both advantages and disadvantages of using Scrum. There are also projects where the effort of one Scrum Team is not enough and some way of handling the scaling needs to be applied.

### 4.7.1 Suitable circumstances

Depending on the context, Scrum could be more or less appropriate to use. According to Williams and Cockburn (2003, p. 40), Scrum is most suited for "**non safety-critical projects with volatile requirements, built by relatively small and skilled collocated teams**". Although projects with several hundred people have been successfully performed using agile development, it is best suited for collocated teams of about 50 people or less in total. It is also suggested that Scrum should not be used for life-critical projects. (Williams & Cockburn 2003)

### 4.7.2 Advantages and disadvantages with Scrum

As in any other framework, there are advantages and disadvantages to Scrum. However, what appears as an advantage can be turned into a disadvantage and vice versa depending on the context. Here are some of the advantages and disadvantages.

#### Advantages

- The short Sprint time facilitates change-management and offers improved control of the project progress.
- Scrum emphasizes the individual through the Daily Scrum and self-organizing teams.
- Backlogs and Burn-down chart gives control over the remaining tasks.
- Scrum focuses on delivering business value and to have a deliverable product at all times.
- Scrum removes the uncertainty of estimating the amount of work far into the future.
- Scrum provides close contact with customer.

#### Disadvantages

- Scrum have been criticized for being unstructured.
- Scrum is not appropriate to use in all circumstances.
- The Scrum Teams are self-organizing, which can be hard in practice.
- Scrum usually opposes company culture and requires a large amount of adaptations.
- Scrum demands a high level of communication, and this can be a burden.
- The Product Owner may oppose the Scrum process and continues to direct the Scrum Team in detail.

4.7.3 Scaling Scrum

Many projects require more than one Scrum Team working simultaneously to get the desired functionality delivered on time. Schwaber (2004) refers to this as a scaled project, where the basic Scrum principles are used in a larger setting. Even if 800 persons are working on a single project, using Scrum they will still be divided into groups of about seven people each. (Schwaber, 2004)

Before all the Scrum Teams starts working in parallel, the facilities, and methods for communication and technology use must be worked out. This is done by one single Scrum Team working from a Backlog that combines non-functional items related to the scaling with items that provide real business value. When the infrastructure needed for multiple Scrum Teams working concurrently is in place, more teams can be added. Schwaber recommends the first team to be divided, with each member forming a team together with new team members. Each of the teams will then start with a Sprint Planning meeting and pick items from the Product Backlog. (Schwaber, 2004)

One commonly used solution to scale the use of Scrum is to conduct a Scrum of Scrums meeting, where one member per Scrum Team participates. The Scrum of Scrums meeting is similar to the Daily Scrum meeting and the questions that all needs to answer is almost the same. Instead of answering the three questions as a person, the participants represent their own Scrum Team as a whole. This meeting especially focuses on areas of overlap and integration involving the different Scrum Teams. If the organization is big, the Scrum of Scrums meeting can be scaled up in recursive manner, which would result in meetings called Scrum of Scrum of Scrums, even though that name is rarely used. (Cohn, 2007)

4.7.4 Disagreement

Scrum has been praised by many organizations and companies, but not everyone is pleased with the increasing adoption rate. There are disagreements about when Scrum should be used and not. Ivar Jacobson, who is an authority on the subject of project management, claims that Scrum is not suited for large organizations and large service oriented systems (Larsson, 2008). Jacobson argues that Scrum is an excellent praxis for managing projects, but that it is not sufficient and additional practices are necessary (Jacobson, 2008). Gustaf Juell-Skielse agrees to Jacobson’s argumentation and says that agile development risk to be “heavy” in complex and large projects with more than 20-30 members (Juell-Skielse, 2007).

4.8 Combining agile development and ISO 9001

As mentioned TL 9000 inherits all the requirements from ISO 9001, so it is a necessary condition to fulfill the ISO 9001 requirements in order to be TL 9000
compliant. There exist several articles about the combination of agile development and ISO 9001.

Nawrocki et al. (2002) was one of the first to consider the idea of combining agile development, in their case XP, with ISO 9001. The aim of their article is to “present a modified version of XP that would be acceptable from the point of view of ISO 9001”. One of the most important problems identified is the lack of written requirements and documentation that all agile development face in one way or another. In this case, it was suggested that the tester should be responsible for writing the documentation, although the overall conclusion in the article is simply that more research is needed. (Nawrocki et al., 2002)

Namioka and Bran (2004) also looked at the same combination, and described a project that passed an ISO audit while using XP. With guidance from ISO experts, a number of ISO-friendly practices for XP are described, such as:

- Requirements are added to the documentation when they are captured, running through the approval process several times if necessary.
- Decisions and plans are documented for example through meeting minutes, and some iterations are used entirely to create documentation.
- Periodic code inspections are performed by engineers outside of the development team.

From this project, the main conclusion is that it is possible to combine ISO and XP. One of the keys to this is to treat documents as any other deliverable and to keep using the feedback for improvements. (Namioka & Bran, 2004)

When it comes to the combination of Scrum and ISO, Vriens (2003) was the first one to look into the issue. A project using Scrum together with XP was successfully certified for ISO 9001 and the details of this project is described in an experience report. Only one ISO process was defined in order to make this work: an abstract level description of how XP and Scrum worked together, with references to books where more detail was needed. The process worked well in the certification, but a number of challenges were highlighted: more senior management involvement is needed, quality assurance processes must be improved, and better tools for risk assessment and management must be developed. (Vriens, 2003)

Wright (2003) argues that to gain ISO certification when using agile development, it is necessary to prove that the process is monitored, measured, and continually improved. This can be done by making documentation, process monitoring and audit trail be created naturally during the development process. The conclusion is that nothing in XP prevents an ISO 9001 certification, and nothing in the ISO audit requirements that reduces agility as long as only natural outputs are used and no artifacts are produced for audit purposes only. (Wright, 2003)

It is also important to remember that the software development process is just a part of the overall company, for which the certification is issued; hence, the quality of the processes before the software part is started is also important to consider. (Wright, 2003)
McMichael and Lombardi (2007) describe a QMS, based on ISO while still maintaining the benefits of being agile. By documenting the already implemented work processes from Scrum, an ISO 9001 aligned system could be created that helped improve training, response to customer enquiries, and the different processes. This in turn led to a decreased number of in-process defects. The procedure descriptions contained the sequence and dependencies of the tasks in a simple, narrative text. However, due to business reasons an ISO certification process was never conducted at the company. (McMichael & Lombardi, 2007)

Jim Schiel (2008) describes how Scrum can be made to work together with ISO 9001 by management involvement and organization policies that are enacted and continuously improved. This is done through a six-step process leading to modifications of the quality policy. First, a quality policy must be created where the commitment of the organization is explained. Internal audits must be in place to ensure policy compliance and the way this is performed must be described. Document control needs to be exercised on formal artifacts, describing storing, versioning, naming etc. These artifacts must also be reviewed and approved by appropriate persons. When a nonconformance is found, there must be a description of how to confirm, prioritize, and assess the impact of the nonconformance. Procedures for preventing and correcting a nonconformance must also be in place. These policies should be updated based on feedback from Sprint Retrospectives and other business realities. (Schiel, 2008)

According to Schiel, all that is needed to ensure quality have to be included in the Definition of Done. All aspects of the definition, for example review and artifact creation, must be reflected in the task breakdown during the Sprint Planning Meeting, and continually followed up during the Sprint. At the end of a Sprint a Sprint Review Record is created, a Sprint Report with the final status of each item and signed approvals. The result of each Retrospective must also be documented so that a quality manager can use the lessons learned to update the quality policies if necessary. (Schiel, 2008)

Stålthane and Hanssen (2008) discuss how it is possible to achieve an ISO 9001 certificate while still using agile development. They base their article on numerous other articles concerning the use of agile development and quality management systems together. In their literature study, Stålthane and Hanssen criticize some of the other authors. Their opinions about the other articles are clearly stated and they comment incorrect conclusions by the authors. By showing critique to other authors and by pointing out their defects, Stålthane and Hanssen gain more credibility in their own research. (Stålthane & Hanssen, 2008)

Stålthane and Hanssen use the same technique as Wright (2003): the ISO 9001 requirements are examined and compared to Scrum, requirement by requirement. Each ISO 9001 requirement that is not clearly conformant with agile development is scrutinized. The reason for the lack of full conformance is identified and possibilities to amend this by extending the development process are described. If a change or adoption is needed in order to fulfill the ISO 9001 requirement, an appropriate one is suggested. Their main conclusions are found to be in line with previous articles on the subject. (Stålthane & Hanssen, 2008)
An ISO 9001 certification audit is not an exact science, since it demands an amount of interpretation by the auditor. This is needed both to interpret the intention of the ISO 9001 requirement and to decide whether the requirement is fulfilled or not. Different auditors perform their own individual interpretation on a case. This may lead to different decisions from auditor to auditor, even though the decision is based on the same facts. To strengthen the validity of their research, two ISO-auditors were consulted to make sure the ISO 9001 requirements were interpreted correctly by the authors. This strengthens the credibility of the article and misunderstandings are reduced. (Stålhane & Hanssen, 2008)

The main difference between agile development and ISO 9001 is the requirements on documentation and in some sense also traceability. Agile development recommends documenting just enough, but if a customer needs more documentation, agile development will not be a hindrance. It is possible to add activities to agile development so that these documents are produced, and still keep the fundamental principles and speed of agile development. There is however a limit to the number of activities and processes that can be added to agile development before it stops being agile. The changes necessary to achieve the ISO 9001 certificate are however well inside those limits. (Stålhane & Hanssen, 2008)

In order to be ISO 9001 compliant, documentation that can be used as proof of conformance and as part of the verification and validation review must be produced. When this is done, there shall be no problem whatsoever to use agile development and keep the ISO 9001 certificate. (Stålhane & Hanssen, 2008)
5 Result

On a corporate level, Motorola is using a waterfall project model to design and develop all of their products. A project team is led and directed by a project manager. Initiated by some individuals, some kind of semi-Scrum was introduced in 2006-2007 to evaluate the possibilities of using agile development. At the time, only one team was working in this way with the project manager also acting as Scrum Master. The daily status meetings and the fact that a Backlog was used was what appeared most Scrum-like.

Before summer 2008, it was decided that Scrum should be widely deployed for all software development in the organization. The employees were divided up in teams of about seven and the first Sprint was started. A Swedish consultant firm specializing in agile development was hired to help with the transition, including general Scrum training with all employees as well as more specific training for Scrum Masters and Product Owners.

5.1 Scrum roles

At Motorola in Linköping, the expected roles from the Scrum literature are all used: the Scrum Team, the Scrum Master, and the Product Owner. Due to the project model in use at a corporate level the additional role of the Product Manager is also described.

5.1.1 The Scrum Team

A Scrum Team is comprised of employees from different fields of expertise, such as testing, hardware abstraction layer, platform, and software architects. It has been pointed out that the personality of the individuals in the team affects the well-being in the group. One member of the team is appointed Scrum Master, educated, and given the associated responsibilities in addition to their normal work.

In some of the teams, certain members have other responsibilities, for example supporting customers and external development teams, leaving less time to work towards the actual Sprint goal. This makes it hard for the teams to estimate the number of items to commit to during a Sprint.

There are also other related problems, such as when a Scrum Team member promises an external part to support them with an errand, but the Product Owner does not prioritize the task high enough to be at the top of the Product Backlog. The helper now have to choose between picking a low priority item instead of a high priority item, or breaking the promise to the external part. A third, and commonly used, option is to fulfill the promise through a private Backlog, bypassing Scrum.

On the other hand, some teams are fundamentalists concerning Scrum and refuse to perform tasks that do not come from the Backlog, which lead to frustration for Project Managers.

It has been discussed to have a team dedicated to respond to urgent tasks only. This way other teams would not be interrupted or disturbed during their Sprint. Today, there is a maintenance team supporting the already released products that also uses Scrum. It works in parallel with the development teams in overlapping Sprints. An
additional support team has been considered. The role of support team could rotate among the teams or it could always be the same team. Still the responsibilities of a support team will require broad skills in the field.

A few teams admit they could follow the Scrum framework more strictly, even though they think they are following it well today.

Generally, the Scrum Teams and their members are satisfied with the working process, which is also reflected in the Fun-index\textsuperscript{7} and the remarkably low number of resignations since Scrum was introduced.

Each team has at least one tester responsible for making sure that the items are completed concerning their functionality. One risk with having testers as a part of the team is that an item that is not completed may be marked as done because the tester feels the pressure from the rest of the Scrum Team to do so.

There is a Definition of Done\textsuperscript{8} that applies to all teams, but one team has made its own additions to achieve a higher level of quality. The same team also writes a Sprint Report when the Sprint is finished, outlining what has been done during the Sprint and the test results.

\section*{5.1.2 The Scrum Master}

Each team has a Scrum Master. The Scrum Masters were given additional training before the adoption of Scrum. The amount of time spent on being a Scrum Master varies largely between the different Scrum Masters, some spend half of their time and others only a small amount of time. One interviewee argues that the Scrum Master role demands hard work and big effort to be used correctly, and that more effort should be put into this. More problem solving and more coaching are also needed from the Scrum Masters. Sometimes it is difficult for the Scrum Master to make the whole team work on one task at a time.

\section*{5.1.3 The Product Owner}

Motorola in Linköping has chosen to have a team of Product Owners instead of having a single Product Owner. The Product Owner-team (known as the PO-team) supervises and updates the Product Backlog. The PO-team prioritizes the items in the Product Backlog and they ask the Scrum Teams to estimate the amount of time it will take to complete the task. The PO-team also answers questions from the Scrum Teams about priorities and questions that are more detailed are handled by Project Managers or other Stakeholders.

The PO-team does not always have a complete view of the item when they create it. They can receive input from the customer, the sales department, the project, and the Scrum Team. Some of the teams encourage the Product Owner to let members of the Scrum Team participate when creating the items.

During the interviews, it is mentioned that there are problems to cooperate within the PO-team. Therefore, it is also difficult to see the PO-team as one unit when they are individually giving different directions to the organization. Nevertheless, the dialogue

\textsuperscript{7} Fun-index is a survey used to evaluate the employees’ mood monthly at Motorola, Linköping.

\textsuperscript{8} Definition of Done is used to explicitly decide when an item is completed or not.
between the PO-team and the Scrum Team is commented as something positive, even though the dialogue could occur more frequently.

It can sometimes be hard for the Product Owner to verify whether an item has full functionality or not on the demonstration. Another problem is the confusing fact that there are both Stakeholders and the PO-team and the developers do not know whom to go to. The Stakeholders are in most cases a Project Manager.

Some of the Product Owners participate in the Retrospective meeting with the Scrum Team to help them evaluate the performed work in the Sprint.

5.1.4 Project Manager
Motorola in Linköping have both Scrum Masters and Project Managers. The role as Project Manager originates from the Project Model in use and not from Scrum. This leads to confusion about where the Project Manager fits into the Scrum framework. It is also mentioned that it is sometimes hard for a Project Manager to get questions answered from the PO-team and the Scrum Team. Suggestions to improve the implementation of Scrum have been made to avoid this problem. One proposal is to let each Product Owner to be responsible for a few teams with one Backlog per Product Owner.

5.2 Practices in Scrum
In addition to the usual Scrum practices, Motorola has added the Scrum of Scrums meeting. This part contains opinions and observations about the Scrum practices, including Scrum of Scrums.

5.2.1 Sprint
A Sprint at Motorola is a three-week period, commencing on a Thursday and ending Tuesday three weeks later, during which development activities are performed. The duration has been chosen to balance between responsiveness to changes and isolation for the teams. Each Sprint starts with a Sprint Planning Meeting and ends with a Sprint Review.

The Wednesday between two Sprints is known as a Slack day and can be freely used by the teams for training, research or other house keeping tasks. Retrospectives are usually held during the Slack day. Some interviewees mention that they sometimes have to use the Slack day to finish tasks that remain from the Sprint.

5.2.2 Sprint Planning Meeting
The Sprint Planning Meeting marks the start of a Sprint when all the Scrum Teams, Product Owners and other stakeholders meet during approximately two hours. It usually opens with information from management and the Product Owners regarding the latest development, customer feedback, and plans for the future. A picture from such an event is shown in Figure 11. Due to the large number of people attending this meeting, it is held in the largest room available.

All items from the Backlog are printed on paper notes that have been sorted according to their priority and attached to a large whiteboard in the meeting room. Included in the notes are a short description of the item, its priority, estimated amount of work needed and the name of the Product Owner responsible for the item.
No public presentation of the top-priority items is usually done, but all the responsible Product Owners are available to provide a more detailed description if any team should need it.

Since the Backlog is common to all the teams, any team may in theory pick any item, as long as the most important items will be picked. However, some teams are working closely with a specific project and their Project Manager. Items regarding these projects are specifically written with the intent to be taken by a certain team and might therefore omit some information that another team would need if they wish to take it.

![Figure 11. Picture from the initial presentation at a Sprint Planning Meeting at Motorola](image)

The teams usually know what the top priority items will be prior to the Sprint Planning Meeting and have discussed which ones they are interested in taking. This is based on the teams’ knowledge about the technical aspects of the tasks. The team also knows the velocity they have achieved during previous Sprints and use that as a guideline for their commitment. Still it might be hard to know how much work a team will be able to commit to, since some team members is occupied with other work and travel arrangements during the Sprint. The teams usually avoid large items that are not time boxed as they include too much uncertainty about the time estimation.

Figure 12 shows the paper notes and the process of picking items from the whiteboard.

![Figure 12. Pictures from Sprint Planning Meeting at Motorola](image)
After the teams have taken all the items they think they will be able to complete during the next Sprint, each is broken down into smaller tasks and the tasks are time estimated. The task breakdown makes it easier to know what time and resources will be needed and to create a plan for the work during the Sprint.

At the end of the meeting, the teams should have taken a number of the items decided to be most important by the Product Owners. In some cases, higher priority items remain while lower priority items have already been taken. The PO-team then usually demands the teams to take the remaining item, with the option to return any item with lower priority that they then would not be able to commit to.

After the Scrum Planning Meeting the teams returns to their team rooms and updates the whiteboard with the new items and tasks.

The Sprint Start Meeting is generally considered a good practice according to the interviewees. It marks the start of something new and shows the things most important to develop at this time.

5.2.3 Daily Scrum
The Daily Scrum is a 15-minute meeting held at a fixed time by each team some time between 9 and 10 every morning. While details differ between teams, the basic structure is pretty much the same: every person gives a summary of what they have done since the last meeting and what they are planned to do today. Impediments are not generally reported. This is because there are no impediments, they have already been addressed outside of the meeting, or the developer has already removed the obstacle himself/herself.

5.2.4 Scrum of Scrums
To gather the combined experiences from all of the Scrum Teams, a special form of Daily Scrum is held three times a week. Scrum of Scrums includes representatives from each of the Scrum Teams, normally their respective Scrum Master, to give updates on what have and will be done during the Sprint. Both technical and Scrum related high-level issues are addressed based on input from the Scrum Teams. Scrum of Scrums is a solution to the scaling problem, to be able to use Scrum in a setting with more than one team.

5.2.5 Demonstration
At the end of each Sprint, a Demo is held. All the teams, Product Owners, stakeholders and other interested gather in the same place as during the Sprint Planning Meeting. The teams then take turns in presenting their items marked as done. Figure 13 shows a picture from a typical Sprint Demonstration at Motorola in Linköping.

Some interviewees have the feeling that items are demonstrated even when they are not completely done, resulting in embarrassing presentations showing limited or broken functionality. Items that do not involve any directly visible functionality, such as structural changes in the architecture or bug fixes, are also hard to demonstrate. Even if the functionality has been implemented as required, the test code showing the
changes might not be written until a later Sprint. A typical demonstration is shown in Figure 13.

5.2.6 Retrospective
At the end of a Sprint each team meet to share their experiences and opinions from the last Sprint. The Retrospective is usually held during the Slack day between two Sprints. Exactly how it is done varies between teams, but commonly a timeline with all the tasks is drawn to refresh the memory of what have happened during the last three weeks. Things that have been positive as well as things that should change are discussed and written on sticky notes. Some teams discuss how important the different proposed changes are to make, and puts the most important on their whiteboard for consideration during the next Sprint. However, the result of the Retrospective is normally not kept in a written form for later reference. It is instead assumed that remaining problems that are not addressed or resolved will reappear in subsequent Retrospectives. One interviewee expresses that this is a waste of information, since nothing guarantees that the same findings will be addressed at a later point in time.

![Figure 13. Picture of a Sprint Demonstration at Motorola](image)

5.2.7 Working environment
All Scrum Teams have their own separate Team Room with whiteboards and workstations. The room and the resources are used in different ways between the teams. Some have adopted pair programming, and their workplaces are therefore designed accordingly. Other teams have their workstations arranged separated from each other. The usage of whiteboards also varies between the teams, some teams using it for both planning and discussion while others do not.

5.2.8 Self-organizing
The interviewees have different opinions of to what extent the Scrum Teams are self-organizing. Some argue that the isolation from unexpected tasks works well and the agreed commitment to items helps the teams to decline additional work during a Sprint.
Other interviewees explain that this only works in theory since some Project Managers bypass the Backlog and directly controls the team during the Sprint. When it comes to tasks related to external customers or suppliers there might well be a related item in the Backlog, but if it does not make it to the top of the list “it has to be done anyway” and is therefore done outside of Scrum. This is because of the fact that the importance of an item might differ between the Product Owner Team, prioritizing the Backlog, and the Product Manager, promising some functionality to the customer. The alternative would be to leave the external part waiting, which is considered worse than the overhead incurred by task switching. The velocity might therefore be very different to the effort the team has actually put in during the Sprint.

Support tasks are also hard to time estimate and plan and may appear at any time during the Sprint. They are therefore something that interferes with the idea of team self-organization.

Opposing this is the argument that no task appears suddenly out of thin air and that it should be possible to include all of them in the Backlog with proper planning.

5.3 Artifacts used in Scrum

Motorola uses three different Backlogs: Product, Sprint, and Release Backlog. Burn-down and Burn-down charts are also used.

5.3.1 Backlog

As mentioned before the Product Backlog is kept up to date by the PO-team and it is written in an excel sheet. The employees generally believe that the Product Backlog is one of the biggest benefits when using Scrum. The members of the PO-team prioritize the Product Backlog together, and range it thereafter. It is a common opinion that the explicit priority is a great advantage, since the developers do not have to make the prioritization themselves. This furthermore gives the developers a better understanding of what is important and not. It has been suggested by the developers to explain the items more clearly on the notes used in the Sprint planning, and then the PO-team would not have to answer so many questions. It has also been discussed to have several Product Backlogs; whether this is a benefit is not agreed. When an item has not been completed during a Sprint, the PO-team marks it as “started not completed”. This is to show that the item has already been worked on, and that there is not as much effort required completing it as the estimate indicates.

The common Product Backlog gives both advantages and disadvantages. An advantage is the increased knowledge distribution within the company. A disadvantage with having one Product Backlog is the difficulty to distribute the items evenly over several teams.

Not all things that must be done are added to the Product Backlog, for example, urgent customer support is not always registered. The Scrum Teams then have to do the prioritization of these unplanned tasks, instead of letting the PO-team do it through the Backlog. The Sprint Backlog consists of all the items a team has committed to during the current Sprint. They are usually displayed on the whiteboard in the team room.
5.3.2 Burn-down chart
The Burn-down chart is drawn on the whiteboard in the team room during the Sprint. All employees are welcome to look at the progress of the Scrum Teams. During the Sprint, the Burn-down chart is made available on a wiki-page on the intranet for anyone that is interested.

5.4 Documentation
According to long time employees, the company does not have a history of extensive documentation. Instead, documents were only produced when necessary and someone would need them. The more formal requirements on documenting the work were introduced when Motorola acquired the company. These are based on the requirements from the currently used waterfall based project model.

Today all the requirements governing software development in the company is found in different SOPs\(^\text{10}\), including the description of what documentation should be written. All of the employees are aware of this, yet very few follow these to their full extent and some have not read them at all.

The reason for this is explained differently between the interviewees. There are different reasons, for example:

- Personal, that it is boring to document or read documentation.
- Organizational, that it is better for new employees to have to understand how things work rather than just copying documented procedures.
- Structural, that the SOP’s include obsolete or unnecessary instructions.

To solve this problem with missing or incomplete documentation, many interviewees suggest these requirements to be included in the Definition of Done so that no item may be marked as done without the proper documentation.

\(^{10}\) SOP is acronym for Standard Operating Procedure
6 Discussion and Analysis

TL 9000 is the QMS used in the telecom industry, based on ISO 9000 and with several additions. If all of the requirements associated with TL 9000 are fulfilled and audited a certification may be awarded to the organization.

6.1 Scrum and quality at Motorola

Scrum is a framework for agile development with the intention of giving faster response to market demands, than traditional waterfall models, while giving the developers tools to increase their performance. Through the ongoing dialogue between developers and customer, facilitated by the Product Owner, Scrum constantly condenses the wants and needs of the customer into requirements that can be converted into functionality. Feedback from customers, through the Product Owner, also helps keeping the product requirements visible, updated, and well adapted to the market needs. The increased involvement given by Scrum is likely to increase the overall quality of the work. It also gives more responsibilities to the Scrum Team to decide how things should be done in practice. This workflow is inherited from the foundation of agile development, the Agile Manifesto, where collaboration, interaction, and working software is identified as more important than documentation, contracts and processes.

The definition of quality used in this Master Thesis is taken directly from the definition found in ISO 9000, and inherited by TL 9000: “the degree to which a set of inherent characteristics fulfils requirements”. A requirement is defined as “need or expectation that is stated, generally implied, or obligatory”. The consequence of this quality definition is that only the level of conformance to a specified set of requirements is important. Subjective aspects, such as the user experience, are not included in the definition if they are not a part of the specified requirements. The requirements must also be defined in some way, or obvious enough to be considered generally implied.

At this point, it is important to underline the difference between product requirements and TL 9000-requirements. On one hand, product requirements are for example the functionality of a product needed by a customer. This is the kind of requirement mentioned in the quality definition. On the other hand, TL 9000-requirements are the prerequisites defined in Requirements Handbook and that needs to be fulfilled by the organization in order to receive a TL 9000-certificate.

In Scrum, each item represents some functionality that is requested by the customer and can therefore be seen as a special form of requirement specification. Each item is prioritized and implemented in a priority-based order. The fulfillment of the product requirements received through an item is therefore explicitly checked, which helps improve the quality through increased requirement fulfillment. The requirements are not always written in specification documents but are more likely found in the description of items or by asking the Product Owner. To combine Scrum with TL 9000 and achieve the desired degree of requirement fulfillment, known as quality, requirements need to be stated in one form or another. The most important step in a study related to the combination of Scrum and TL 9000 would be to see if the requirements imposed by the QMS are followed. It is commonly believed that ISO
9000, and therefore TL 9000, requires large amounts of documentation so any incompatibility concerning this is important to note.

### 6.1.1 Scrum and Total Quality Management

Scrum can also be compared to Total Quality Management (TQM). Six cornerstones are identified in the management theory that can all be identified to various degrees in Scrum.

The customer is put in the center in Scrum through the facilitation of customer input done by the Product Owner. Acting as the link between the customer and the development team, the Product Owner constantly works to ensure that the delivered software is what the customer expects.

In Scrum, decisions are made as late as possible, preferring short term Sprint planning over long-term project plans. The decisions should be made by the persons with as much information as possible about the situation. For example, the Scrum team is using its knowledge about its own performance and progress to make estimations and planning the work.

While Scrum does not think in terms of processes as pre-defined sets of steps that should be followed regardless of the circumstances, it is still a framework with guidelines that needs to be followed. The Scrum framework may in this sense be seen as number of interconnected processes. Customer and supplier input is also highly valued in the framework.

Quality improvements are a key component of the Scrum framework, through the constant adjustment of the methods in use and evaluation of the work done. In practice, this is done at several places in the framework. For example, the Daily Scrum facilitates the presentation of the approach currently in use every day, giving direct feedback on how things are going. The high level of communication between team members at all times is also important in this improvement process. At the end of every sprint, the evaluation done in the Sprint Retrospective also forms a base for improvements of the approach.

Conditions for involvement is created in Scrum through self managed teams, focusing on the individuals, and adjustments to the work based on feedback. According to the TQM-theory, this will help increased motivation and improve the results from the employees.

Dedicated Leadership is something that is outside of the scope of the Scrum framework. The idea of inspiring and motivating the co-workers is however very present in the framework.

From this follows that the Scrum framework is in line with the principal features of TQM and nothing is hindering the quality concept from being successfully applied. What remains to see is whether the way Scrum is used is consistent with this and that all specific features from the TL 9000 requirements are fulfilled.
6.2 Differences between the Scrum framework and Scrum at Motorola

When looking at how Motorola in Linköping is using Scrum and comparing it with how Scrum is explained in the literature a few differences are revealed. This is a natural consequence of using a framework in a complex reality, where business value and other aspects have to be considered. Some of these differences are minor when viewed from a quality perspective, since they do not affect the quality in any way. Others may have a direct impact on the quality, either positive or negative.

6.2.1 Common Backlog

One of the most obvious features in the use of Scrum at Motorola is the common Backlog for eight teams, containing items from several different customer projects. However, the literature does not explicitly specify how the Backlog should be handled when working on multiple projects. From a quality perspective, the important aspect is that requirements are specified, implemented, and verified. There is no reason why this cannot be done with a common Backlog, as long as the Product Owner fulfills the responsibilities associated with the role.

Some of the items in the Product Backlog are intentionally written to be picked by a specific team. This is more or less like a separate Backlog for that team, even though the team specific items are also included in the common Backlog. The Scrum framework does not specify the relationship between the Backlog and a team, other than that there should exist a Backlog that a team can pick from. This particular use of the common Backlog is therefore not fundamentally different.

6.2.2 Product Owner-team

Another obvious difference between the Scrum framework and Scrum at Motorola is that the role of Product Owner is replaced by a team of several Product Owners. This might introduce a number of problems related to the distribution of information, since it is nearly impossible for the different Product Owners to keep up to date with all aspects of the products and Backlog items. The reason that the Scrum framework only has a single Product Owner is to avoid contradicting information. A single Product Owner is also a single point of contact, while a team introduces the difficulty of knowing whom to ask a certain question. This is in fact a real problem at Motorola and efforts are made to avoid it in the future. The PO-team has daily meetings where they share information and discuss the user stories and questions from the Scrum Teams. This meeting introduces a delay since sometimes no answers can be given to a specific question before this meeting. The creation of a PO-team is at the same time a solution to the scaling problem. One person simply cannot complete all the tasks to manage a Backlog for multiple projects.

When considering the impact on quality the chosen approach brings both advantages and disadvantages. On the positive side, a Product Owner-team has more time to communicate with customers and developers, manage the Backlog, and spot potential problems. The negative side is that if the responsibilities are not strictly defined, important work may be overlooked and forgotten. It is therefore very important to make sure that when using a Product Owner-team the role of the team members are well defined and obvious to the rest of the organization.
6.2.3 Sprint Planning
At the start of a Sprint Planning Meeting, the top priority items should be explained in detail by the Product Owner, so that the team knows the expectations for each item. This presentation is not done at Motorola, but the Product Owners are instead available to clarify and answer questions about the items. The items are also available for inspection the day before the start of a Sprint and at the estimation-planning meeting, which gives more time for the Scrum Teams to discuss and ask for clarifications. Since the idea of the presentation is to make sure the team understands what needs to be done, this should be considered an equally good practice from a quality perspective.

6.2.4 During the Sprint
The Case study shows that not all Scrum Teams are following the Scrum framework strictly at all times. Not every team has Daily Scrum every day, something that contradicts Scrum and may cause a decrease in the exchange of information within the Scrum Team. Skipping the Daily Scrum occasionally will probably not have a major impact, but it may in the end lead to negative results. Therefore, it is recommended to always have Daily Scrum.

At the Daily Scrum, the question about impediments is rarely answered. This is one of the three questions all Scrum Team members should answer during the Daily Scrum. While this does not directly prevent high quality, but the Scrum Master could help the team to work more efficiently if they reported impediments at the Daily Scrum more often. Most teams do not update their Burn-down chart at the Scrum board every day, something that the Scrum framework suggests. The Burn-down chart is used to visualize the project advance and give the team a hint of remaining work for the Sprint. Hence, this should not be a problem concerning quality. It is recommended for the teams to follow the Scrum framework as strictly as possible.

Some work is for various reasons never added to the Backlog at all. This may be different tasks such as supporting customers and other developers, or work that a project manager needs to have done immediately outside of the Sprint commitments. While this helps in keeping things running smoothly from a management perspective, it may also contradict the intentions of Scrum by breaking the Sprint isolation and increasing the pressure on team members to choose between Sprint items and other work. From a quality perspective, it may also decrease the level of traceability when work is done without specified requirements and verification. This needs to be addressed if the quality could be affected in a negative way, while keeping in mind that it is not possible to add every single task in the organization to the Backlog. It is the responsibility of the Scrum Master to guard the team from tasks that not originate from the teams own Sprint Backlog.

6.2.5 Roles and responsibilities
The Scrum Masters at Motorola does not put as much effort in being a Scrum Master as the Scrum framework recommends. This is mentioned during the interviews, but also noted in the observations. It is recommended to adjust the role of the Scrum Master so that more effort is put on, for example customer collaboration and education of the Scrum Team. This of course implies that the Scrum Master puts less effort in software development.
Scrum says it is important to involve the testers directly in the development. Motorola have chosen to have the testers as members of the Scrum Teams. It has been mentioned that the testers may pass tasks that not are done because that leads to progress and higher velocity for their own team. If this happens, it might result in lower code-quality and in the end a less good product. In order to support the tester, a more specific definition of done is recommended to be adopted. One team has already extended the global Definition of Done with their own additions, and inspiration could be collected from that specific team. If a new improved Definition of Done is introduced and used, testers’ membership should not be a problem in regards to quality.

6.2.6 Scrum of Scrums
The Scrum of Scrums meeting is used to share knowledge between teams and it is part of the scaling solution. It is a Motorola-specific solution and not described in any of the literature. At the Scrum of Scrums, all teams share helpful information through the Scrum Masters, while giving the Product Owners better insight into the work. From a quality perspective, this should be considered as a positive addition.

6.2.7 At the end of the Sprint
When a Sprint has come to its end, a demonstration is held where all teams demonstrate their result and experiences from the Sprint. Scrum calls this gathering for Sprint Review and some details differ slightly from how Motorola performs it. Besides using different words, the criteria for presenting the Sprint results are different at Motorola compared to the Scrum framework. Even if a team has not completed an item, it is presented and demonstrated at the demo. This contradicts the Scrum framework, which claims an item must fulfill the Definition of Done in order to be demonstrated. To make sure all teams have reached the agreed Definition of Done, one of the Product Owners should take time and discuss with the Scrum Team before each item is approved, and allowed to be presented at the demonstration. Not having a complete Sprint review, also removes the dialogue between stakeholders, for example customers, through the Product Owner, and the Scrum Team if the implementation meets the expectations.

When the Sprint is finished one of the teams write a Sprint Report. It includes, for example, the Sprint goal, the description of the work done during the Sprint, and the most important changes from the Retrospective. A Sprint Report increases traceability and facilitates the quality improvement process. This is also a good way to document experiences from the Sprint. Based on the given reasons, the quality could be increased by having all teams write a Sprint Report after each Sprint.

From a quality viewpoint, this pose a problem since the Product Owner may think that everything that is demonstrated is completely done. Either the individuals or the team believes the item is done and everything is working while it is not, or they think it will easily be finished during the Slack day. If a problem is found after the item has been demonstrated and marked as done, it will probably be inconvenient for the team to tell the Product Owner it is no longer done and put the item back into the Product Backlog. The team might try to fix it outside of Scrum, or ignore the fact that it is not completed. As a result, the requirements cannot be traced and verified. In order to achieve a high level of quality, the Definition of done must be well specified and met for all demonstrated items.
Not all teams have a Sprint Retrospective meeting after every Sprint and the results from the Retrospective are not always retained. The Retrospective is held to evaluate the Sprint and to summarize the experiences earned from the Sprint. Without the Retrospective, these important experiences will not be discussed, and the same problems may have to be solved again later. It is recommended to always have the Retrospective meeting after a Sprint. Scrum does not say much about retaining the Retrospective result, but in order to achieve the TL 9000-certificate doing so could be an advantage.

6.2.8 Customer collaboration and Release Management

The Scrum framework recommends a high level of customer collaboration, for example it is recommended to let the customer participate at the Sprint Review. This is not the case at Motorola. To fulfill this recommendation, a higher amount of customer collaboration must be established. A higher level of customer collaboration also leads to customer satisfaction.

At Motorola, some customer releases are made occasionally on the request from the customer. The Scrum framework demands releases after each Sprint. Having releases after every Sprint facilitates customer feedback since the customer has the possibility to give feedback and see the result of the feedback already after next Sprint. Therefore, it is recommended for Motorola to make new releases after every Sprint.

6.2.9 Conclusions about the use of Scrum at Motorola

The conclusion is that there are some minor differences between the Scrum framework and Scrum at Motorola. The following differences and suggestions for improvements have been identified:

- The roles of the Product Owner-team members should be well defined so that all responsibilities are clearly covered.
- Work should be specified and included in the Backlog, if omission of requirements specifications and verification, risk decreasing the quality.
- The Scrum framework is recommended to be followed as strictly as possible by the Scrum Team, especially:
  - Hold a Sprint Retrospective after a Sprint.
  - Conduct Daily Scrum every day
  - Report impediments to the Scrum Master at Daily Scrum
  - Update Burn-down chart
- Scrum Teams should not be disturbed during a Sprint with tasks that does not originate from the Sprint Backlog.
- The Scrum Master role should be less involved with software development and more involved with organizational issues.
- Scrum of Scrum increases the sharing of knowledge between teams.
- Adoption of a more specific Definition of Done is recommended.
- A Sprint Review, where the Product Owner and Scrum Team discuss the implementation, should be held before an item is considered done and allowed to be demonstrated.
- Sprint Reports may help to share the experiences between teams.
- An increase of the level of customer collaboration is recommended.
- Make new releases after every Sprint.
As mentioned earlier there is a difference between how Scrum is explained in the literature and how Motorola has implemented it. The difference is small, but important. However, if the above stated recommendations are followed the difference will be without significance in a quality aspect. Then the Scrum framework and Scrum at Motorola are similar enough to be seen as the same.

6.3 Combining ISO 9000 and Scrum

To receive a TL 9000-certification all the associated requirements must be fulfilled and audited. Since the ISO 9000-requirements are a subset of the TL 9000-requirements, literature dealing with the combination of ISO and agile development, specifically Scrum, is used as a stepping-stone. Even if the literature discusses different approaches to combining ISO 9000 and agile development – Scrum, XP and a more generic form – the conclusions are all still valid on a general level in the context of TL 9000 and Scrum.

One thing that is worth repeating is that quality auditing is not an exact science, but involves a fair deal of interpretation by the auditor. Consequently, the result of this discussion will only be recommendations what is likely to be important aspects to consider when combining Scrum and TL 9000. It is also important to remember that the software development is only a part of the work in the organization, and that the overall quality depends upon the quality of the surrounding processes. The responsibility of fulfilling the requirements is therefore not restricted only to Scrum, but may also depend on other parts of the organization.

The aim of both the QMS and agile thinking is in the end very similar: to deliver what the customer wants and to continually improve the organization. What differs is the underlying philosophy and how this is done in practice.

From an ISO 9000 perspective the organization needs to demonstrate that a process is monitored, measured, and continually improved to be able to receive a certification. To do this, a quality policy must be in place. It should include the organizational policies regarding internal audits, document control on artifacts, review, and approval of artifacts, and handling of any nonconformance.

As several authors point out, the main difference between ISO 9000 and agile development is the requirements on documentation. While ISO 9000 prefers documents to be produced as proof of conformance and for verification and validation of requirements, Agile thinking instead underlines the importance of communication and the ability to adapt to a changing reality.

It is however not fair to say that these approaches are direct opposite of each other and cannot be combined. The literature shows that it is possible to add activities to Scrum so that the necessary documents are created. One suggestion is to treat documentation as any other deliverable. In practice, this could mean that items for writing documentation have to be included in the Product Backlog as any other item in the Backlog. At the same time, as Wright (2003) argues, no such artifacts should be produced for audit purposes only, as this creates overhead. There is also a limit on how many additional activities that may be added to the framework before its agility is severely reduced.
Instead, the documentation requirements could be included as a part of the Definition of done, since it should include all activities needed to ensure the quality. To document what has actually been done, a Sprint Report containing the achievements and problems encountered should be created at the end of each Sprint. This document should be approved, signed, and saved based on the document control policy.

The suggestion that testers should be responsible for writing the documentation is probably attractive for the developers, but would at the same time be hard to use in practice due to the distribution of knowledge.

Contrary to the documentation issue, continual improvements are something that both ISO 9000 and Scrum have in common. The feedback that is gathered through different sources, such as the Daily Scrum and Sprint Retrospective, should be used as much as possible to adapt and improve processes and tools. To facilitate learning between teams and use the knowledge to improve any involved processes it is recommended to document the Sprint Retrospective.

From the literature, the conclusion can be drawn that Scrum and ISO 9000 could be combined if the following recommended features are present:

- A quality policy must be in place, used and continually improved based on feedback.
- Documentation tasks should be a part of the Definition of Done so that it always is carried out consistently.
- A Sprint Report should be created at the end of every Sprint to summarize what was done and the problems encountered.
- Continual improvements are made based on feedback generated from different activities, for example a document created based on the Sprint Retrospective.

### 6.4 Scrum at Motorola compared to the requirements of TL 9000

Scrum is already shown to be compliant with ISO 9001 in chapter 6.3. What remains is to determine whether Scrum can fulfill the requirements from TL 9000. The TL 9000-requirements has already been discussed in chapter 3.2.4, and the requirements concerning only hardware and services are excluded since hardware is not discussed in this Master Thesis. In this section, each requirement will be compared to Scrum at Motorola and labeled depending on how they are treated in Scrum or if they are out of the scope.

If the requirement is marked as “Fulfilled”, it means that Scrum satisfies the requirement and there should be no problem, caused by this requirement, to achieve the TL 9000 certificate.

If the requirement is not fulfilled by Scrum and is within the scope of the Master thesis, a modification is needed. These requirements will be marked “Needs modification” since a change is
needed to fulfill the requirement. It is not a modification of Scrum that is suggested, but rather an addition to the work process.

If the requirement is marked as “Contradictory”, the requirement states the opposite of Scrum. In this case, Scrum at Motorola needs to be changed, since the TL 9000 requirements are not changeable.

If the requirement is marked as “Out of scope”, the requirement is outside the frames for this Master Thesis. The scope in this Master Thesis is only treating requirements operating on a level where Scrum works, which is within software development on a team level. If the requirement is marked with “Out of scope” the requirement is not operating on this level, and is hence a requirement that should be handled by another process, for example XP or Lean Development.

The fact that there is a limit to how many processes it is possible to add to Scrum, has been mentioned previously. As Stålhane and Hanssen argue, there is a limit to how many new artifacts that can be added to the work process without loosing the agile concept. The intent is to minimize the amount of additional processes and yet be TL 9000 compliant. When the additional processes are formed, a discussion will be held whether Scrum at Motorola still will be agile when these processes are included.

### 6.4.1 Summary of the comparison

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.3.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>5.2.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>5.2.C.2</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>5.4.1.C.1</td>
<td>Needs modification</td>
</tr>
<tr>
<td>5.4.2.C.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>5.4.2.C.2</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>5.4.2.C.3</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>5.5.3.C.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>6.2.2.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>6.2.2.C.2</td>
<td>Needs modification</td>
</tr>
<tr>
<td>6.2.2.C.3</td>
<td>Out of scope</td>
</tr>
<tr>
<td>6.2.2.C.4</td>
<td>Out of scope</td>
</tr>
<tr>
<td>6.2.2.C.5</td>
<td>Out of scope</td>
</tr>
<tr>
<td>6.2.2.C.6</td>
<td>Out of scope</td>
</tr>
<tr>
<td>6.3.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>6.4.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.1.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.1.C.2</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.1.C.3</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.1.C.4</td>
<td>Needs modification</td>
</tr>
<tr>
<td>Section</td>
<td>Status</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------</td>
</tr>
<tr>
<td>7.1.HS.1</td>
<td>Needs modification</td>
</tr>
<tr>
<td>7.2.2.C.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>7.2.2.C.2</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>7.2.3.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.2.3.C.2</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.2.3.C.3</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.2.3.HS.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.2.3.HS.2</td>
<td>Needs modification</td>
</tr>
<tr>
<td>7.3.1.C.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>7.3.1.C.2</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>7.3.1.C.3</td>
<td>Needs modification</td>
</tr>
<tr>
<td>7.3.1.HS.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.3.1.HS.2</td>
<td>Needs modification</td>
</tr>
<tr>
<td>7.3.1.S.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.3.1.S.2</td>
<td>Estimation</td>
</tr>
<tr>
<td>7.3.1.S.3</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.3.1.S.4</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.3.2.C.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>7.3.2.C.2</td>
<td>Needs modification</td>
</tr>
<tr>
<td>7.3.2.C.3</td>
<td>Needs modification</td>
</tr>
<tr>
<td>7.3.2.S.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>7.3.3.HS.1</td>
<td>Needs modification</td>
</tr>
<tr>
<td>7.3.5.C.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>7.3.5.HS.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.3.5.HS.2</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.3.5.S.1</td>
<td>Needs modification</td>
</tr>
<tr>
<td>7.3.6.S.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>7.3.7.C.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>7.3.7.C.2</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>7.3.7.C.3</td>
<td>Needs modification</td>
</tr>
<tr>
<td>7.4.1.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.5.1.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.5.1.C.2</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.5.1.HS.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.5.1.HS.2</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.5.1.S.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.5.1.S.2</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.5.1.S.3</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.5.3.HS.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.5.5.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>7.5.5.S.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>8.2.1.C.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>8.2.3.C.1</td>
<td>Needs modification</td>
</tr>
<tr>
<td>8.2.4.S.1</td>
<td>Needs modification</td>
</tr>
<tr>
<td>8.4.C.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>8.4.HS.1</td>
<td>Out of scope</td>
</tr>
<tr>
<td>8.5.1.C.1</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>8.5.1.C.2</td>
<td>Fulfilled</td>
</tr>
<tr>
<td>8.5.2.S.1</td>
<td>Out of scope</td>
</tr>
</tbody>
</table>
6.4.2 Comparison of the TL 9000 requirements and Scrum at Motorola

**Requirement [4.2.3.C.1] Control of Customer-Supplied Documents and Data**
It is not the role of Scrum to establish or maintain a documented procedure for customer-supplied data. Following it might be so, but not establishing it.

*Out of scope*

**Requirement [5.2.C.1] Customer Relationship Development**
The top management is not involved in the Scrum Process; hence, this is not a requirement for Scrum to deal with.

*Out of scope*

Scrum recommends a high level of customer collaboration, with all customers, and if this recommendation is followed, requirement 5.2.C.2 is not a problem. The high level of customer collaboration should also raise the number of satisfied customers.

*Fulfilled*

**Requirement [5.4.1.C.1] Quality Objectives**
This requirement includes the whole Measurements handbook. Measurements handbook consists of a set of measures to measure progress and evaluate results of quality management implementation. The requirement is to have the measures implemented not for the measures to be within certain limits. To fulfill this requirement the measures must be implemented, if they already are implemented this requirement is fulfilled. The implementation of these measures does not necessarily involve Scrum, and therefore this requirement is out of scope.

*Needs modification*

**Requirement [5.4.2.C.1] Long- and Short-Term Quality Planning**
The quality planning at Motorola is divided into Sprints, which can be seen as the short-term quality planning. The Sprint goal can be considered the goals for improving quality and customer satisfaction. The items at the Scrum board contain relevant business factors for the organization and its customers and they are established jointly with the customer. The performance to these goals will be monitored and reported at the Sprint Demonstration. For this requirement to be fulfilled, a long-term goal for quality planning must be set. Setting a long-term goal, longer than a Sprint, is out of scope since this does not concern Scrum. Therefore, this requirement will be marked as Fulfilled.

*Fulfilled*
Requirement [5.4.2.C.2] Customer Input
Scrum uses short iterations, Sprints, and after each Sprint a new release is done. The customer gives feedback on the new release and it is possible to make changes to the next release. This way, Scrum works close to the customer and the customer’s opinion can fast be transformed into changes. Scrum also encourages development together with the customer.

Fulfilled

Requirement [5.4.2.C.3] Supplier Input
Suppliers are stakeholders in the project and are therefore represented by the Product Owner. Supplier input is used continually during the development.

Fulfilled

Requirement [5.5.3.C.1] Organization Performance Feedback
The quality performance of the employees will, if necessary, be discussed during the Retrospective meeting. How the organization informs the employees of the customer satisfaction is not for Scrum to determine.

Fulfilled

Requirement [6.2.2.C.1] Internal course development
Internal courses is out of scope for Scrum, but if an internal Scrum course would like to be held, it must follow methods to ensure consistency according to this requirement.

Out of scope

Requirement [6.2.2.C.2] Quality and Process Improvement Concepts
The key components from this requirement are addressed by the Scrum framework, so it is important that it is being followed. The Sprint Retrospective is used to provide feedback for improvement of quality and processes. It is the role of the Scrum Master to make sure the teams are properly educated in the fundamental concepts of continual improvement, problem solving and customer satisfaction. This demands the Scrum Master to put more effort into organizational issues, like customer collaboration and education of the Scrum Team. By doing that the awareness of quality aspects will increase and in the end lead to better products.

Needs modification
Requirement [6.2.2.C.3] Product Quality Training Opportunity Awareness
Training and education is outside the scope of Scrum.

Requirement [6.2.2.C.4] Electrostatic Discharge Training
Training and education is outside the scope of Scrum.

Requirement [6.2.2.C.5] Advanced Quality Training
Training and education is outside the scope of Scrum.

Requirement [6.2.2.C.6] Hazardous Conditions Training Content
Training and education is outside the scope of Scrum.

Requirement [6.3.C.1] Infrastructure
Infrastructure is not related to Scrum and this requirement is therefore not in the scope of this Master Thesis.

Requirement [6.4.C.1] Work Areas
This requirement concerns storage and handling of hardware and is therefore out of scope.

Requirement [7.1.C.1] Life Cycle Model
A life cycle that covers the entire life of its products needs to be established and maintained. While Scrum is an important part of this, it is outside of the entire life cycle is outside of the scope.
Requirement [7.1.C.2] Disaster Recovery
Documented plans for disaster recovery should be established and maintained, something that is completely outside the scope.

There have to be a documented procedure for the actions taken when a product or service is discontinued. This is mainly handled at a higher organization level than Scrum, with some details that could possibly be related to Scrum.

One of the details is how any future support issues are handled. This could be done either by some kind of support team outside of Scrum, or by adding an item to the Backlog. Another detail is how documentation and software are archived, something that have to be handled by the Scrum Team during the development. The actual documentation is controlled by other requirements.

As a whole, this requirement is out of scope for the Scrum framework.

Internally developed software and tools needs to be subject to appropriate quality methods. The note to this requirement mentions design and testing tools together with the software used to build the product as examples. In practice, everything developed to be used in the production of a product must be subject to the same quality expectations as the product itself. This is not directly related to the implementation of the Scrum framework, but still related to what is done in the Scrum Teams and needs further consideration.

Requirement [7.1.HS.1] Configuration Management Plan
According to the TL 9000-dictionary, configuration management is defined as:

“A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report changes, processing and implementation status, and verify conformance to specified requirements.” (QuEST Forum, 2006b, p. Glossary - 1)

This requirement states that a plan for configuration management must exist and include a number of different topics. The scope of these activities has to be identified and a schedule for performing them must be presented. Any tools, methods, and documented procedures that are used must also be described and included in this document. In addition to this the document must describe any involved organizations.
and their responsibilities, the required level of control for each item and the point at which items are brought under configuration management.

In Scrum, this direction and surveillance of a configuration item comes from the Product Backlog items and Sprint Reviews. The items contain requirements that are at the very latest verified at the Sprint Review. What might be needed is closer tracking of what happens to an item through its life. The Configuration Management Plan must also include descriptions of how Scrum works and is used in the organization. Therefore, some slight changes in the work process are needed in order to fulfill this requirement.

**Needs modification**

**Requirement [7.2.2.C.1] Closure Tracking**
The TL 9000 requirement 7.2.2, inherited from ISO 9000, calls for a review of the requirements before the organization commit itself to supply the specified product. It must make sure there are defined product requirements, no requirements changes are left unresolved, and that the organization is be able to meet the requirements. The requirement [7.2.2.C.1] states that any action that is the result of such a review has to be tracked to closure.

In Scrum, the requirements are defined together with the customer when the Product Backlog items are written by stakeholders and Product Owners. At the Sprint start, a team may choose to commit to the item only if they think they are able to solve the problem. During the Sprint, changes to the requirements are not allowed but the Product Owner may need to clarify certain aspects of the item to the team. Items can be tracked until they are finally marked as done. This means that this requirement is fulfilled.

**Fulfilled**

**Requirement [7.2.2.C.2] Contract Review**
A contract review process should be in place in the organization, including nine different requisites. By looking at them one by one, it is possible to see if they are compatible with Scrum.

The first one mandates a product acceptance and criteria review process. This is an integral part in Scrum, where the requirements are refined, and the product updated until it is accepted by the customer.

The second requisite mandates a method to handled problems after product acceptance. Either this could be handled completely outside of Scrum, by a separate support team, or by adding the changes needed to the Product Backlog. In the later case, the same process is repeated as was previously used to refine the product until the state when it was accepted.

The third requisite mandates a plan for removal and correction of nonconformities after warranty or during the contracted maintenance period. Like in the previous
requisite, this could either be handled by adding an item to the Product Backlog and let the Scrum Team handle it, or have a separate organization for late corrections.

The fourth requisite mandates that risks and possible contingencies have to be identified. It is always hard to know when all risks have been identified, but Scrum takes the approach to deal with risks by constantly giving the opportunity to change most parameters of the development. The identification of risks is therefore integrated in the Scrum process. Risks with the entire project have to be handled outside of Scrum.

The fifth requisite mandates proprietary to be adequately protected. The sixth requisite mandates definition of the organization’s responsibility with regard to outsourced work. Neither is handled by Scrum.

The seventh requisite mandates the customer to be included in the requirements, specifications, and acceptance activity. This is a natural part of Scrum through the active customer involvement both through the Product Owner, Product Backlog items and at Sprint Reviews.

The seventh and eight requisite mandates customer supplied facilities, tools, and software to be reviewed and referenced standards and procedures to be included. This is not, and should not be, handled by Scrum.

Seeing that the requisites are either fulfilled or handled outside of Scrum this requirement is seen as fulfilled.

Fulfilled

Requirement [7.2.3.C.1] Notification About Problems
There must exist procedures for how customers are informed about service affecting problems. This is not related to the Scrum framework.

Out of scope

Requirement [7.2.3.C.2] Problem Severity
The response of the organization to a customer-reported problem is guided by a severity level, which is assigned outside of the Scrum framework.

Out of scope

Requirement [7.2.3.C.3] Problem Escalation
Procedures to escalate customer-reported problems should exist, which is outside of the Scrum framework.

Out of scope
Requirement [7.2.3.HS.1] Organization’s Recall Process
A process for identifying and recalling products unfit to remain in service must exist, which is outside of the Scrum framework.

Out of scope

Requirement [7.2.3.HS.2] Design and Development Process Quality Measurements Data Reporting
Communications shall include reporting and evaluation of a jointly agreed set of design and development process measurements if the customer requests it. This requirement is only active if the customer requests the measurements. If the customer does so, additional processes are needed to fulfill this requirement. The measurements shall be jointly agreed with the customer, this authorize Motorola to control which measurements to perform. It is important that the measurements do not interfere with the agile concept.

Needs modification

Requirement [7.3.1.C.1] Project Plan
A project plan must include information about the project and be based on the product life cycle model. The project organization must be described along with a number of other items, such as the project team and its roles, responsibilities and accountabilities. The methods used in the development must be described, even though a reference to the life cycle model is sufficient, as well as the customer, user and supplier involvement.

Nothing in this requirement contradicts the Scrum framework, but it needs to be written based on the way Scrum is actually used.

Fulfilled

Requirement [7.3.1.C.2] Requirements Traceability
The organization is required to establish and maintain a method to trace documented requirements through design and test. In Scrum, the product requirements are given by the items in the Product Backlog. This requires that all actions must pass through the Product Backlog as items. If they do so, they can easily be traced. The Definition of Done is a documented method to make sure the requirements are fulfilled and the Sprint Review can be seen as the finish that ties it all together.

Fulfilled
Requirement [7.3.1.C.3] Test Planning
This requirement demands a documented test plan. Since the testers are part of the Scrum Teams, tests are performed within the Scrum framework. A test plan must therefore exist for all tests, which have to include all the requirements from 7.3.1.C.3.

Needs modification

Requirement [7.3.1.HS.1] Migration Planning
Migration between operation environments for products or system is not handled by Scrum.

Out of scope

Requirement [7.3.1.HS.2] Design and Development Process Quality Measurement Planning and Implementation
Measurements are crucial to TL 9000, but are not affecting Scrum directly.

Out of scope

Requirement [7.3.1.S.1] Integration Planning
When integrating software components a plan is needed to ensure that the different parts will work together. The plan must include how the integration is done, the different responsibilities, the schedule, and the test requirements. With Scrum, this integration is done continually during each Sprint and this needs to be described. When the integration process performed in the end of each Sprint is described in an integration plan this requirement should be fulfilled. The integration plan must include all subsets from this requirement.

Needs modification

Requirement [7.3.1.S.2] Estimation
To control a project, a method by which project factors can be estimated must exist. In a related note, project factors are described to include product size, complexity, requirements changes, effort, staffing, schedules, cost, quality, reliability, and productivity. Some of these requisites are out of scope for Scrum, staffing and cost. Staffing since it is an activity outside of the project and cost because the cost is set before the project start. Estimation is a natural part of Scrum since it is based on estimating items. Product size, complexity, requirements changes, effort, schedules, quality, reliability, and productivity are all handled within Scrum, in different ways.

Fulfilled
Requirement [7.3.1.S.3] Computer Resources
The critical resources of the target computer, on which the developed software is intended to be run on, have to be estimated and tracked. This requirement is highly relevant to software development, but is not handled on a Scrum level but rather at a lower level.

Requirement [7.3.1.S.4] Regression Test Planning
There must be a plan for regression testing. This is handled outside of Scrum.

Requirement [7.3.2.C.1] Customer and Supplier Input
This requirement states that there have to be methods for gathering and using input from customers and suppliers during the development. This is what Scrum is making sure through the role of the Product Owner, who keeps a constant dialogue with the customer and converts their input and ideas into prioritized Product Backlog items.

Requirement [7.3.2.C.2] Design and Development Requirements
All design and development requirements on the product must be documented, and a number of requisites need to be fulfilled. The documentation must include such things as quality, safety, and reliability requirements, functions and capabilities of the product, and design constraints.

Some of these product requirements are found in the Product Backlog items, for example the desired function and capabilities of the product. However, some of the requested product requirements are not documented in Scrum, and thus changes are needed.

Requirement [7.3.2.C.3] Requirements Allocation
This requirement commands the developers to document how the software is built. Scrum does not fulfill this requirement, but an additional process that requires appropriate documentation of the software architecture would do so. This could for example be done in a Sprint Report that the teams writes after each Sprint.

Requirement [7.3.2.S.1] Identification of Software Requirements
It is required to determine, analyze, and document the requirements for the software components of the system. In Scrum, this is performed by the Product Owner who has essential knowledge about the customer requests. The Product Owner analyzes
the customer needs and transforms them into items. The items are documented in the Product Backlog by the Product Owner. This way Scrum fulfills the requirement.

**Fulfilled**

**Requirement [7.3.3.HS.1] Design and Development Output**

It is natural for the developers to produce source code, it is their main assignment. Before any source code is produced the systems architecture must be designed. This is partly performed through the items and how they are divided from each other, but also in task breakdown. System detailed design is set on a lower level than where Scrum operates and is therefore out of scope. Scrum tries to aim at as little documentation as possible, but a certain amount of documentation is always needed. The user documentation should be produced as items through the Product Backlog. This demands additional processes but the other three are, as indicated, already fulfilled.

**Needs modification**

**Requirement [7.3.5.C.1] Verification of Documentation**

Before any documentation is sent to customer or user, it shall be controlled. If documentation is performed as items through the Product Backlog, they are controlled by the Definition of Done before they are marked as ready. This way documentation is controlled before it is sent to customer.

**Fulfilled**

**Requirement [7.3.5.HS.1] Stress Testing**

Stress testing of a product is likely to be done as a part of testing in the Scrum Team, but is not handled by Scrum.

**Out of scope**

**Requirement [7.3.5.HS.2] Abnormal Conditions**

Testing of a product under abnormal conditions is likely to be done as a part of testing in the Scrum Team, but is not handled by Scrum.

**Out of scope**
Requirement [7.3.5.S.1] System Testing
The software has to be subject to a system test according to the documented test plan before being released. Due to the complex nature of a system test, this cannot usually be done during a single Sprint. Instead, it is handled by dividing the system test into multiple items. These items are taken by different Scrum Teams until the system is ready to be released. In this way, Scrum fulfils the requirements.

Fulfilled

Requirement [7.3.6.S.1] Release Management
This requirement states that there must be a defined method for how software is released and delivered together with the related documentation. The method must include release planning information before the actual release, schedules for product introduction and release, detailed descriptions of new features and changes, and advisories on changes to contractual term.

This is not fulfilled since a release is currently not prepared after each Sprint, which would be needed to fulfill this requirement. In that case, the release-planning schedule is simple to communicate, since every third week a new release is ready to be delivered. The new features and changes to the software are also easy to distinguish from the start of the Sprint, when a Scrum Team has committed to an item and promised the requested functionality to be delivered at the end of the Sprint. Contractual terms are not handled inside of Scrum. To fulfill this requirement, a modification of the use of Scrum is needed.

Needs modification

Requirement [7.3.7.C.1] Change Management Process
The organization must have procedures for handling changes of requirements and design at any time during the product life cycle. If anything is changed that adversely affect agreed conditions a review must be held with the customer before approval. The procedures must include impact analysis, planning, implementation, testing, documentation, communication, and review and approval.

When using Scrum during the product development, all of this is handled through the use of the Product Backlog items and Sprint Review. The Product Owner decides, with input from the customer and other stakeholders, which functional changes should be made and in what order. The Scrum Team will then commit to the top priority items and work to implement this during the Sprint. Impact analysis, planning, implementation, testing, and documentation is done during the Sprint. At the Sprint Review, the Product Owner will be able to review and decide if the new functionality meets the expectations or if more work is needed.

Changes at a later point in the product life cycle are handled outside of Scrum. Therefore, nothing in this requirement contradicts the Scrum framework.

Fulfilled
Requirement [7.3.7.C.2] Informing Customers
Customers should be informed about changes to contractual commitments based on design conditions. These may either be inside of Scrum, in which case there is an ongoing dialogue between the customer and the Scrum Team through the Product Owner, or outside of Scrum. Nothing in this requirement contradicts Scrum.

**Fulfilled**

Requirement [7.3.7.C.3] Problem Resolution Configuration Management
The configuration management system must track fixes to problems and make sure they are used in future versions of the software. This requirement is not completely within the Scrum framework, but for those parts that are additions are needed. More specifically, the way Scrum is currently used does not cater for the tracking of fixes. It could for example be done by documenting the changes that fixes problems made during a Sprint. If fixes to problems are developed outside of Scrum those fixes needs to be handled separately.

**Needs modification**

Requirement [7.4.1.C.1] Purchasing Procedure(s)
The purchasing of products has to follow a documented procedure. This is outside the scope of Scrum.

**Out of scope**

Requirement [7.5.1.C.1] Service Resources
Customer contact employees must be trained and equipped with the necessary tools and resources, which is outside the scope of Scrum.

**Out of scope**

Requirement [7.5.1.C.2] Product Delivery
There must be methods to minimize the interference at the customer during delivery and installation. This is not handled by Scrum.

**Out of scope**

Requirement [7.5.1.HS.1] Emergency Service
Service and resource availability during an emergency is outside the scope of Scrum.

**Out of scope**
Requirement [7.5.1.HS.2] Installation Plan
Establishing and maintaining installation plans are outside the scope of Scrum.

Out of scope

Requirement [7.5.1.S.1] Patching Procedure(s)
A documented patching procedure must be in place and used in the organization. It must guide the decision to use patching, describe how patches are developed, and used, be consistent with what the customer and contracts demands, and ensure that a statement of the impact of a patch is given to the customer.

If a Scrum Team is asked to solve a problem and decides the best way to do it is by making a patch, it will need to follow these procedures. However, the creation of the patching procedures is not related to the Scrum framework and is therefore outside the scope.

Out of scope

Requirement [7.5.1.S.2] Patch Documentation
There must be a method which ensures that all the documentation required to describe, test, install, and apply a patch is verified and delivered to the customer together with the patch. Like in the case of [7.5.1.S.1], these methods are needed when a Scrum Team decides to solve a problem by patching, but outside the scope of Scrum to create.

Out of scope

Requirement [7.5.1.S.3] Replication
Data replication has to follow documented procedures, which is outside the scope of Scrum.

Out of scope

Requirement [7.5.3.HS.1] Product Identification
Each product must be identified based on its model and version, which is outside the scope of Scrum.

Out of scope

Requirement [7.5.5.C.1] Electrostatic Discharge Sensitive (ESDS) Protection
Anti-static protection must be used where applicable, which is outside the scope of Scrum.

Out of scope
Requirement [7.5.5.HS.1] Packaging and Labeling Verification
Packaging and labeling must conform to specified requirements, which is outside the scope of Scrum.

Out of scope

Requirement [7.5.5.S.1] Software Virus Protection
Software virus prevention, detection, and removal must be used, which is outside the scope of Scrum.

Out of scope

Requirement [8.2.1.C.1] Customer Satisfaction
Scrum constantly uses Sprints and new releases are made after every Sprint. The customer gives feedback on the new release and it is possible to make changes to the next release. This can be seen as a method for collecting customer satisfaction data and customer feedback.

Fulfilled

Requirement [8.2.3.C.1] Process Measurement
When appropriate, process measurements shall be identified, documented, and monitored. This is not performed in Scrum and an additional process must hence be established in order to achieve the TL 9000 certificate.

Needs modification

Requirement [8.2.4.S.1] Test Documentation
Scrum demands as little documentation as possible and TL 9000 demands testers to document the test results. In a way, this could be seen as contradicting, but by adding a documentation processes to Scrum this requirement could be fulfilled without neglecting the Scrum framework. The documentation process is performed by the testers and could be included in the Definition of Done.

Needs modification

Requirement [8.4.C.1] Trend Analysis of Nonconforming Product
Making a trend analysis is outside the scope of Scrum.

Out of scope
Requirement [8.4.HS.1] Field Performance Data
Collecting field performance data is also outside the scope of Scrum.

Out of scope

Requirement [8.5.1.C.1] Continual Improvement Program(s)
The Retrospective meeting is a continual improvement program for all processes within the organization. More specifically, Scrum demands a high level of customer collaboration, and customer collaboration result in satisfied customers. Having a detailed definition of done together with Sprint Review can be seen as quality and reliability improvement programs for the product.

Fulfilled

Requirement [8.5.1.C.2] Employee Participation
The Scrum Retrospective is a forum for discussing improvements of the team and the organization. The most important conclusions from Retrospective will be brought up on the Scrum of Scrums Retrospective and this way the whole organization will continually improve.

Fulfilled

Requirement [8.5.2.S.1] Problem Resolution
Bug correction is not handled by Scrum. This is handled on a lower level than where Scrum is operating, for example XP.

Out of scope

6.5 Changes needed to be compliant with TL 9000
When comparing the TL 9000-requirements to Scrum, a number of non-compliances are found. These have been divided into different categories, depending on what part or artifact they relate to. In order for Motorola to be TL 9000-compliant, the following additions to Scrum are suggested.

6.5.1 Definition of Done
The global Definition of Done gives general minimum requirements that must be completed in order to mark one item as done. The definition should be used as a help during the task breakdown to make sure all relevant tasks are created and later completed.

Actions that should be taken on all items are added to the global Definition of Done. Based on the TL 9000-requirements, some additions to the Definition of Done should be made.
The definition must contain the necessary level of quality assurance, such as testing in accordance with test plans and documentation. A test plan must be created that fulfills [7.3.1.C.3]. This includes the scope of testing, types of tests, and methods of reporting and resolving defects. The result of the testing and the following actions must also be recorded.

It is also suggested that creation of documentation for product requirements should be included in the definition. This includes a number of recommended design and development requirements from [7.3.2.C.2] together with design constraints and computer resources for the target computer. Verification of what have actually been created for the item is also a very important part.

By using a global definition of done that includes the above mentioned, the testing and documentation will always be completed before an item is marked as done. At the end of the Sprint, the item is demonstrated at the Sprint Review, where the Product Owner will ensure that the Definition of Done is fulfilled. If not, the item is returned to the Backlog to be completed during a future Sprint. Another advantage of a detailed definition of done is that it can be a part of the quality and reliability improvement programs for the product.

6.5.2 Sprint Report
In order to fulfill some of the TL 9000 requirements it is recommended to write a Sprint Report. The Sprint Report should be written after each Sprint by the team and should not be extensive documentation. The Retrospective meeting gives a good basis for the Sprint Report and the most relevant findings should be documented here.

The Sprint Report should also contain information about the items from the Sprint Backlog and what has happened to them, for example, if the item has been worked on before, if it is done or if any changes have been performed to it. A description of how the Software Architecture is built is an essential part of the Sprint Report and should be included. The solutions and changes made to fix problems during the Sprint should also be documented in the Sprint Report in order to be able to track fixes in the future, but also to share the knowledge.

6.5.3 Product Backlog
All documentation that is not written in the Sprint Report or in the Definition of Done, for example user documentation, shall be performed as items through the Product Backlog. This way the documentation will be quality controlled when it is inspected in accordance with the Definition of Done.

Each item shall, beside the regular specifications, contain all the requirements of the software component, in order to fulfill the requirements of TL 9000. By letting the documentation be produced as items through the Backlog, traceability is increased, since activities that are not performed through the Product Backlog, will not be possible to trace. It also has other advantages. If all activities are handled through the Product Backlog, the teams will not be disturbed in their work during the Sprint, which emphasizes self-organization.

The Product Backlog is handled and prioritized by the Product Owner-team by updating a shared Excel-file. This is a simple and efficient way to solve the problem,
but if not handled carefully it could result in conflicting versions of the Backlog. This is something that the Scrum framework is specifically critical towards. The traceability of a specific item – who added it, changed it, deleted it etc – is also very low when a primitive type of database is used. The Product Backlog may be thought of as containing the product specifications. In that case, it should be handled in a more structured way. Instead of the shared Excel-file, a more suitable alternative is recommended.

6.5.4 Release
The intention of Scrum is that at the end of every Sprint, a potentially releasable product will be available and ready for the customer to use. This gives very direct feedback from the customer of to what degree the developed software meets the requirements. While it is not always practically possible to have a releasable product after each Sprint, due to factors such as the complexity of the product or the number of Scrum Teams working in parallel, a release should never be far away. This could for example mean that a release would be possible after a limited number of Sprints containing items related to the integration, testing, and release.

The customer would still be able to give feedback on a product that has not been away from development for a long time, while still giving time to perform crucial system integration and testing. For this, an integration plan that describes what needs to be done in order to release the product must be created.

6.5.5 Measurements
TL 9000 requires a number of measurements to be identified and included in the different processes. Requirement [5.4.1.C.1] includes the measurements from the Measurements Handbook, [7.2.3.HS.2] gives the customer the opportunity to define measurements, and [8.2.3.C.1] requires process measurements to be identified, documented, and monitored at appropriate points. Some documentation concerning measurement do already exist, inspiration and input may be gathered here.

It is not possible to give general advice about where these measurements should be inserted into different processes. One important thing to remember though is the importance of keeping the agility of the Scrum framework. Too many or misplaced measurements will impede the progress of the development.

6.5.6 Scrum
The role of the Scrum Master has several possibilities for improvements concerning organizational issues. It is desirable for the Scrum Master to collaborate more with the customer in order to achieve a higher customer satisfaction. The Scrum Master also could focus more on removing impediments and guarding the Scrum Team from external forces. It is furthermore the role of the Scrum Master to make sure the teams are properly educated in the fundamental concepts of continual improvement, problem solving and customer satisfaction.

In order to fulfill the TL 9000 requirements the use of Scrum should be documented. The plan for configuration management shall contain a description of how Scrum is performed in the organization. Some documentation of the use of Scrum at Motorola do already exist, inspiration and input may be gathered here.
6.5.7 Supplier input
The developers will most likely have the prime need of supplier input, since the suppliers are most often third-party software components that need to be integrated. Since supply chain input is important in software development, it is recommended to apply a method for soliciting and using the supplier input.

6.5.8 Summary
A summary of the recommended additions:

- A global Definition of Done should be established.
- A Sprint Report should be written by each team at the end of each Sprint.
- Creation of any documentation not written in a Sprint Report or as a result of the Definition of Done should be handled as an item in the Product Backlog.
- Each item in the Product Backlog should include the requirements of the software component.
- The Product Backlog should be handled in a more suitable way than through a shared Excel-file.
- A release should never be further away than a small number of Sprints for integration and testing.
- Measurements should be implemented as required by the TL 9000 while trying to ensure the agility of Scrum.
- The role of the Scrum Master should be enhanced by focusing more on the Scrum principles.
- The way Scrum is used in the organization should be documented.
- Supplier input should be used in the development.

It is important to note that these are only recommended additions.
7 Conclusions

This chapter contains the conclusions and final considerations of the Master Thesis. The Problem Definition stated in 1.3 Problem definition will be answered as well as the Aim 1.2. The Problem definition will be stated in *italics* below and the answers are written in regular.

*The master thesis aims to answer the following questions:*

**How could Motorola meet the TL 9000 requirements by using Scrum?**
- What is quality management?
- What are the requirements of TL 9000?
- How is Scrum explained in existing literature?
- How does Motorola use Scrum?

The four sub-questions have already been answered in chapter 3 and 4, and they are the foundation to answer the overall question.

The answer to the overall question is: Yes, Motorola can meet the TL 9000 requirements by using Scrum, under the condition that some additional processes are performed and that other parts of the organization also fulfills the remaining requirements. This is needed since there are requirements that are out of scope for the Scrum framework.

Some changes and additions need to be made to the usage of the Product Backlog:
- Work should be specified and included in the backlog, if omission of requirements specifications and verification, risk decreasing the quality.
- Each item should include all the requirements of the software component.
- The Product Backlog should be handled in a more suitable way than through a shared Excel-file.
- Creation of any documentation not written in a Sprint Report or as a result of the Definition of Done should be handled as an item in the Product Backlog.

The Definition of Done should also be used in a different way:
- Adoption of a more specific global Definition of Done is recommended, to ensure that documentation and testing are always carried out consistently on items.

The roles of the Scrum Master the Product Owner-team should be clarified:
- The Scrum Master role should be less involved with software development and more involved with organizational issues.
- The roles of the Product Owner-team members should be well defined so that all responsibilities are clearly covered.

Some changes directly related to quality improvements should also be made:
- A Sprint Report should be written by each team at the end of each Sprint, to help share experiences between teams.
- Continual improvements should be made based on feedback generated in different activities.
A quality policy must be in place, used and continually improved based on feedback.
Measurements should be implemented as required by TL 9000 while trying to maintain the agility of Scrum.

The Scrum framework should also be followed more closely and additional practices should be made in accordance with the specific needs of the organization:
- The Scrum framework is recommended to be followed as strictly as possible by the Scrum Team, especially:
  - Hold a Sprint Retrospective after a Sprint.
  - Conduct Daily Scrum every day
  - Report impediments to the Scrum Master at Daily Scrum
  - Update Burn-down chart
- Scrum Teams should not be disturbed during a Sprint with tasks that does not originate from the Sprint Backlog.
- A Sprint Review, where the Product Owner and Scrum Team discuss the implementation, should be held before an item is considered done and allowed to be demonstrated.
- A release should never be further away than a small number of Sprints for integration and testing, preferably after each Sprint.
- The way Scrum is used in the organization should be documented.
- Supplier input should be used in the development.
- An increase in the level of customer collaboration is recommended.

It is also important to underline that these additions are recommendations, not requirements.

**What are the requirements of TL 9000 that Scrum contradicts or does not deal with?**
There are no requirements in TL 9000 that clearly contradicts Scrum. It is an obvious advantage that no requirements are contradicting Scrum, because of that no major changes are needed in Scrum. There are however some requirements that are not dealt with in Scrum.

**Is it possible to make additions to Scrum so that it includes these requirements?**
Yes. The requirements that are within the level of where Scrum operates are fulfilled by additional processes. The other requirements have not been discussed further, and have to be considered by other parts of the organization in order to be fulfilled. This is however not within the scope of this Master Thesis.
8 Final considerations

After presenting the conclusion of the thesis, some remarks are necessary to make about the application and implementation of the result. Topics for further research are also suggested.

8.1 Validity of the conclusion

It is important to point out that auditing is not an exact science. The decision of whether a requirement is fulfilled or not may vary between auditors, and it could therefore be hard to verify if the suggested additions are sufficient. There is also the possibility that the TL 9000 requirements have been incorrectly interpreted due to the complex nature of the specifications and the certification process. The conclusions should therefore be seen as suggestions that may help achieving a certification.

Between the time when the observations and interviews were performed and the conclusions are presented, some changes to the practices have already been made at Motorola in Linköping. In some cases, the circumstances that the suggestions are based on have also changed. However, based on the findings of the case study, the conclusion is valid.

8.2 Maintaining the Agility

When adding supplementary processes to the Scrum framework in order to fulfill TL 9000 requirements, there is a risk of diminishing the agility and flexibility of the framework. As mentioned before, it is not possible to add an unlimited number of processes to agile development and still call it agile. The agility of the agile development is gradually decreased with every additional process added. Some requirements may also decrease the agility more than others may. To determine whether the Scrum framework together with the additions is still agile is a complex task. The changes may need to be applied to the organization before the effect on agility can be evaluated.

After considering the suggested changes, we believe that the reduction in agility is not of a major nature and the advantages however outweigh the disadvantages. The extended Scrum framework should still be considered agile. In order not to decrease the agility even more, no more processes should be added.

8.3 Future research

This thesis only focuses on a narrow part of the intersection between agile development and quality theory. Related to this, a number of interesting topics for further research can be identified.

The idea behind Scrum was first described in 1986, and the framework has been used widely in the commercial world for the last few years. With the increased adoption, the combination of Scrum and traditional project models is an issue that Motorola and other companies are currently facing. This could be looked into together with the question if there is any recent research that could improve the Scrum framework.
If the result from this thesis were implemented at some point in the future, a study of the practical implementation and the certification process would be interesting. The result would show if the suggested modifications actually result in a TL 9000 certification. The achieved quality could also be compared to a traditional project model or the Scrum framework without the additional processes. In that way, the suggested combination of Scrum and TL 9000 could be evaluated in practice.
Bibliography
Printed sources


Additional sources


### Appendix A: Principles behind the Agile Manifesto

<table>
<thead>
<tr>
<th>Principles behind the Agile Manifesto</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</td>
</tr>
<tr>
<td>- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.</td>
</tr>
<tr>
<td>- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
</tr>
<tr>
<td>- Business people and developers must work together daily throughout the project.</td>
</tr>
<tr>
<td>- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</td>
</tr>
<tr>
<td>- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</td>
</tr>
<tr>
<td>- Working software is the primary measure of progress.</td>
</tr>
<tr>
<td>- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</td>
</tr>
<tr>
<td>- Continuous attention to technical excellence and good design enhances agility.</td>
</tr>
<tr>
<td>- Simplicity--the art of maximizing the amount of work not done--is essential.</td>
</tr>
<tr>
<td>- The best architectures, requirements, and designs emerge from self-organizing teams.</td>
</tr>
<tr>
<td>- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.</td>
</tr>
</tbody>
</table>

(Agile Alliance, 2001b)
Appendix B: Interview questions

1. For how long have you been working at Motorola in Linköping, and what are your position, education, and previous experience?

2. What tasks are your responsibilities?

3. For how long have you been working with Scrum?

4. Do you think you have a good understanding of the concept of Scrum?

5. Could you describe how the Scrum-process works at Motorola in Linköping?

6. What do you think is working well with Scrum?

7. What do you think are not working well with Scrum?

8. What tasks are you occupied with that is not Scrum related?

9. What is done in opposite to the Scrum philosophy?

10. What guidelines for documentation of the work are in place today? Where do they originate?