Valtteri Konttinen

Towards Disciplined Software Development

ISO 9001:2008 Based Software Process Improvement in SME

Helsinki Metropolia University of Applied Sciences

Master's Degree

Information Technology

Master's Thesis

31 May 2016

Preface

I would like to thank my supervisors Erja Nikunen and Mika Lankila for their support and guidance. Special thanks to Erja for your endless patience and flexibility and to Mika for always keeping my feet on the ground.

I would also like to thank my mother for letting me use her study and for sustaining my nourishment levels when the little misses at my house didn't agree on my work schedule.

I am eternally grateful to my lovely wife Jenny and beautiful daughters Vera and Alva for allowing me the time to complete my studies. I know it hasn't been easy for you. Thank you for all your sacrifices. There is no way I could have pulled this of without you guys.

This thesis is dedicated to the memory of my father who suddenly passed away last summer. Thank you for everything Faija!

Porvoo, 31 May 2016
Valtteri Konttinen

Author	Valtteri Konttinen
Title	Towards Disciplined Software Development: ISO 9001:2008 Based Software Process Improvement in SME
Number of Pages Date	58 pages + 16 appendices 31 May 2016
Degree	Master of Engineering
Degree Programme	Information Technology
Specialisation option	Mobile Programming
Instructor(s)	Mika Lankila, Chief Financial Officer Erja Nikunen, Principal Lecturer

The case company's quality management system has recently been awarded the ISO 9001:2008 certificate. The next necessary step is to incorporate the software processes into the company's quality management system. The goal of this study was to identify the key software process improvement areas based the ISO 9001:2008 requirements and to provide recommended actions for improvement.

Sufficient background knowledge on quality in general and quality in the context of SMEs and software was gained and ISO 9001:2008 and its application in the context of software was studied. Based on the gained knowledge methods and models for assessing the current software processes and deriving the recommended improvement actions were selected and developed. A self-assessment was performed and respective improvement actions were derived and presented.

The self-assessment results indicated that the software processes would need to be redesigned from the ground up. Hence, recommendations targeted on key software process areas could not be derived. Alternatively, an example process based on the Disciplined Agile Delivery framework was presented for the company to consider when the software process is redesigned.

Even though the results of the assessment were not what was expected the study itself provides a sound basis for software process improvement efforts in the future. Plenty of valuable knowledge and experience on software process improvement was gained which will to serve the case company well in the future.

Keywords	Quality, Software process, SME, Software quality, Software process improvement, ISO 9001:2008

Tekijä	Valtteri Konttinen
Otsikko	Kohti kurinalaista ohjelmistokehitystä: ISO 9001:2008 standardiin pohjautuva ohjelmistoprosessien parantaminen PK-yrityksessä
Sivumäärä Aika	58 sivua + 16 liitettä 31 Toukokuu 2016
Tutkinto	Insinööri, YAMK
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Mobiiliohjelmointi
Ohjaaja(t)	Mika Lankila, Talousjohtaja Erja Nikunen, Yliopettaja

Tutkimuksen kohteena olevan yrityksen laatujärjestelmälle on hiljattain myönnetty ISO 9001:2008 sertifikaatti. Seuraava tärkeä vaihe on liittää yrityksen ohjelmistokehitysprosessit osaksi laatujärjestelmää. Tämän tutkimuksen tavoitteena oli tunnistaa ohjelmistokehityksen avainprosessit ISO 9001:2008 vaatimuksiin pohjautuen ja esittää parannusehdotuksia.

Ensivaiheessa hankittiin tarvittavat taustatiedot laadusta yleisesti ja laadusta ohjelmistokehityksen ja PK-yritysten kontekstissa sekä ISO 9001:2008 standardista ja sen soveltamisesta ohjelmistokehitykseen. Taustatietojen pohjalta valittiin sekä laadittiin tarvittavat menetelmät ja mallit nykyisen ohjelmistokehitysprosessin arvioimiseksi ja parannusehdotusten johtamiseksi. Ohjelmistokehitysprosessin itsearviointi suoritettiin ja tulosten pohjalta esitettiin parannusehdotuksia.

Itsearvioinnin tulokset osoittivat, että ohjelmistokehitysprosessi tulee suunnitella uudelleen alusta alkaen. Tästä syystä ohjelmistokehitysprosessin avain alueisiin kohdennettuja parannusehdotuksia ei pystytty johtamaan ja esittämään. Vaihtoehtoisesti esitettiin Disciplined Agile Delivery –kehykseen pohjautuva esimerkki prosessi, jota yrityksen tulisi harkita suunnitellessaan uutta ohjelmistokehitysprosessia.

Vaikka suoritetun itsearvioinnin tulokset eivät olleet odotusten mukaisia, itse tutkimus tarjoaa kestävän pohjan yrityksen ohjelmistokehitysprosessien kehittämiselle tulevaisuudessa. Tutkimuksen aikana kartutettiin paljon arvokasta tietotaitoa sekä kokemusta yrityksen tulevaisuuden tarpeisiin.

Avainsanat	Laatu, Ohjelmistoprosessi, PK-yritys, Ohjelmiston laatu, Ohjelmistokehitysprosessin parantaminen, ISO 9001:2008

Contents

Table of Contents

List of figures

List of tables

Abbreviations and acronyms

QMS	Quality Management System
SPI	Software Process Improvement
SME	Small and Medium Sized Enterprises
TQM	Total Quality Management
SDLC	Software Development Life Cycle
AQL	Accepted Quality Level
QMP	Quality Management Principle
PDCA	Plan-Do-Check-Act cycle or methodology
PIMS	Profit Impact of Market Strategy
RPQ	Relative Perceived Quality
ROI	Return of Investment
SQC	Statistical Quality Control
EC	European Commission
EU	European Union
OECD	Organization for Economic Co-operation and Development
PRM	Process Reference Model
VSE	Very Small Entity
PAM	Process Assessment Model
DP	Deployment Package
PM	Project Management
SI	Software Implementation
SDLC	Software Development Lifecycle
DAD	Disciplined Agile Delivery
QA	Quality Assurance

1 Introduction

This research was conducted for a small sized security technology company operating in Espoo Finland. The case company is a solution provider company that handles the needs of its customers from concept to production and maintenance. The company's strategy is to utilize industry state-of-the-art hardware and software components and build highly customized solutions out of them by integrating them together with in-house developed software modules.

The company's key customer solutions are based on an in-house developed software platform. These software related solutions cover ~30% of the company's revenues while approximately 10% of its employees are doing mainly software related work. It is recognized that the role of software capabilities is critical for the company to gain and maintain its competitive edge. Firstly, a larger amount of the company's revenue is known to be somehow dependent on the in-house software development and maintenance activities, however, the exact indirect impact is difficult to estimate. Secondly, the agile software capabilities are key to the company's competitive edge, that is the capability to rapidly build and deliver highly customized solutions to the company's customers. Thus, the role of software capabilities is seen increasingly critical for the company's business.

1.1 Business Problem

To maintain the company's competitive advantage and to push the company forward, the company's management has decided to invest in QMS (Quality Management System) development. The company's management has selected the ISO 9001:2008 QMS, which is the most prevalent QMS standard in the world, as the basis for the QMS development. The reasons for a company to apply ISO 9001:2008 are multiple. In brief, it results in improved customer satisfaction and enhanced quality of the entire organization (Robitaille, 2010). Furthermore, ISO 9001:2008 certification is increasingly seen as a requirement for a supplier to even participate in tendering (Oskarsson, 1999).

The case company has recently been awarded the ISO 9001:2008 certificate for its QMS and ISO 14001 for the environment related processes. However, given the com-

pany's current structure and revenue composition, emphasis in QMS development has been on other than software processes, namely sales and maintenance. This was not due to undervaluing the importance of the company's software capabilities, but mainly a question of resources and priority as maintenance is critical for the end-customer and thus, for the company's business model. The company's QMS development has now progressed to a point where the next necessary step is to incorporate and improve the company's software processes.

1.2 Purpose and Outcome

The purpose of this study is to assess the case company's current software processes against the ISO 9001:2008 requirements and to identify the key improvement areas. Furthermore, based on the findings, to recommend SPI (*Software Process Improvement*) actions that the company can take to improve its processes. The outcome of this study is a list of key SPI areas with recommended SPI actions for the case company to take. Figure 1 below highlights the focus of this study.

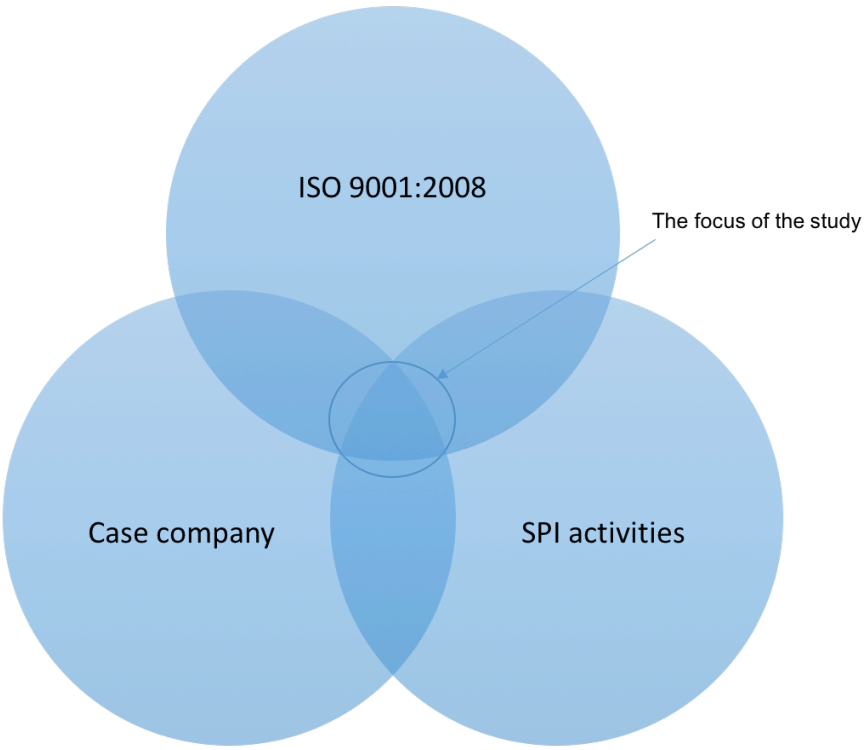

Figure 1. The focus of this study.

Figure 1 above elicits how the special characteristics of the case company, ISO 9001:2008 requirements and SPI activities are interrelated and affect each other. In the epicentre are the key improvement areas based on ISO 9001:2008 requirements and the case company with respective SPI activities.

This research attempts to answer the following question: “Based on ISO 9001:2008 and the current state of the case company, what are the key software process areas for improvement and how to improve?”. Answering this question includes addressing the following sub-questions:

- What is the current state of the software processes in the case company?
- What are the key software processes and activities that the case company’s improvement efforts should be primarily directed to?
- For the key improvement areas, what are some of the possible SPI actions to take?

This study is structured in 9 sections. In Section 1 the context, objectives and key concepts of the study are introduced. Section 2 introduces the research process and research methodology and data collection techniques. In Section 3 quality is discussed in general. Section 4 further discusses quality in the context of SMEs (*Small and Medium Sized Enterprises*), and finally, in Section 5 quality is discussed in the context of software. Section 6 introduces ISO 9001:2008 on a detailed level. Section 7 sets the scope for the SPI efforts taken in the following section. In Section 8 the development and application of the models for identifying the key SPI areas and gaps of the current process are explained, and furthermore, the models are applied, the results are analysed and the SPI recommendations are derived and presented. Finally, the study is critically discussed and conclusions are drawn in Section 9.

1.3 Key Concepts

In this section the key concepts of the study are briefly introduced. Each concept is discussed in its own sub-section and definitions for their meaning in the context of the present study are given.

1.3.1 Quality

Juran, one of the leading quality gurus and top contributors to TQM (*Total Quality Management*) and modern quality thinking, summarized his definition of quality in (Juran, 1988) as “fitness for use”, which depicts the philosophy behind modern quality thinking and TQM (Bath, 2010). TQM and ISO 9001 both recognize that customer expectations are a key concept of quality, quality is about understanding and meeting, or even exceeding, customer’s current and future expectations (Dale, et al., 2013; International Organization for Standardization, 2008a). In this study quality refers to the modern rendition of quality, one promoted by TQM and ISO 9001.

1.3.2 Software Process

ISO gives the following definition for a process “An activity or set of activities using resources, and managed in order to enable the transformation of inputs in to outputs, can be considered as a process” (International Organization for Standardization, 2008a). Although the definition by ISO 9001 is generally adequate it lacks the software specific aspects. Further, the CMM (Software Engineering Institute, Carnegie Mellon University, 1993) gives the following, more software specific, definition for a software process “A set of activities, methods, and practises that people use to develop and maintain software and the associated products”. Even though the definition by the CMM is from 1993, more than 20 years ago, it is still valid and adequate today. Hence, in this study software process refers to the definition given by the CMM.

It should be noted that in this study the term software process is intentionally used, and not software development process, since the term software process covers all software related processes from development to maintenance. Furthermore, it should be noted that in this study the term software process never refers to any specific SDLC (*Software Development Life Cycle*) models, such as waterfall or agile.

1.3.3 SME

According to a book by Robitaille (Robitaille, 2010) there is no exact definition for SME. In some literature a distinction is made between very small, small and medium sized organizations. However, in the context of this study a rigorous definition is not neces-

sary. It suffices to understand that small organizations in general tend to share common characteristics different from their larger counterparts (Robitaille, 2010).

Even though this study is conducted for an organization of less than 50 employees, from which ~10% do software related work (the focus of this study), most of the discussions on special characteristics of SMEs are valid in the context of any SME. Whereas, a clear distinction is made when the discussion explicitly refers to software SMEs.

1.4 Summary

In this section the case company and the business problem were introduced. Furthermore, the purpose of the study was discussed and definitions for the key concepts of this study were given. The argumentation and background to support these definitions was intentionally kept brief, since each of these concepts is discussed in more detail in the later sections.

2 Research Process and Methodology

In this section the research process and methodology are briefly introduced. The step-by-step research process is being presented and discussed, and further, the employed research strategy and data collection methods are introduced.

2.1 Research Process

In this section the research process is introduced. The step-by-step research process is presented in Table 1 below. For each step the table briefly illustrates the objective and the planned approach for achieving it.

Table 1. The step-by-step research process.

Step	Objective	Approach
1	Gain an understanding on what quality is.	Find and read suitable sources on quality and quality management.
2	Gain an understanding on special characteristics of SMEs with respect to quality.	Find and read suitable sources on quality management in SMEs. Build on the knowledge gained in the previous step by incorporating the knowledge gained on SMEs.
3	Gain an understanding on special characteristics of software with respect to quality.	Find and read suitable sources on software quality management and SPI frameworks. Build on the knowledge gained in the previous steps by incorporating the knowledge gained on software quality management and SPI frameworks.
4	Gain an understanding on ISO 9001:2008 with respect to software quality.	Thoroughly study ISO 9001:2008 standard and related software standards, guides and technical reports. Find and read suitable sources on applying ISO 9001 in the context of software. Build on the knowledge gained in the previous steps by incorporating the knowledge gained on ISO 9001 and relate.
5	Gain an understanding on the current state/maturity of the case company's software processes and identify the key improvements areas.	Data about the current software processes and practises is collected by participating in multiple roles to the software development and maintenance activities of the case company. The data is analysed with respect to ISO 9001:2008 requirements by conducting a self-assessment. A suitable method for performing a self-assessment and identifying the key process areas for SPI is selected/developed based on the background knowledge gained in the previous steps.
6	Develop a set of recommended SPI actions for the case company.	Recommended SPI actions for the company are developed based on ISO 9001:2008 requirements, the current state of the case company's software processes and the background knowledge gained in previous steps. A suitable method for identifying the shortcomings of the current software processes and deriving the SPI recommendations is selected/developed based on the background knowledge gained in the previous steps.

Table 1 above shows the detailed step-by-step research process. The research starts with a comprehensive study of background information on quality and quality management in general and in the context of SMEs and software. Furthermore, ISO 9001:2008 and its application to software is studied comprehensively. The conducted theoretical background section tries to answer questions such as “What is quality?” and “What are some special characteristics of SMEs and software with respect to quality?”.

After the theoretical background, the current state of the case company’s software processes is analysed. The data is collected and methods for software process assessment and analysis are discussed. The method for self-assessment is introduced here as well as the model for identifying the key process areas for improvement. The purpose was to develop and introduce a model for performing a self-assessment on the case company’s software processes that is based on ISO 9001:2008 requirements and special characteristics of the company. Finally, the application of the model is reported and the results analysed.

Recommended SPI actions for the case company are then developed based on the self-assessment results. Methods for deriving the SPI recommendations are discussed and the model for performing the gap analysis and to derive the recommendations is introduced. The purpose was to develop and introduce a method/model for deriving SPI recommendations based on ISO 9001:2008 requirements and the results of the self-assessment. Finally, the application of the model is described and the recommended SPI actions are presented.

2.2 Research Approach

According to a guide by Oxford University (Oxford University Press, no date) qualitative research is used when the research question requires an understanding of processes. Instead of relying on numerical data the data is based on knowledge of individuals and social groups in the natural setting. Qualitative data is recorded as text as opposed to the numerical data employed in quantitative research. Hence, the research strategy employed in this study is Qualitative field study.

The primary data collection method in the present study was participant observation, with the emphasis on participating. Informal conversations are also a critical component of participant observation and are largely employed in this study. Participant ob-

servation is natural choice given the author's experience working with the case company's software activities for a number of years in multiple roles.

The field notes are documented and embedded directly into the self-assessment template in Appendix 1. A snippet of the template is provided in Table 2 below, as an example.

Table 2. An example of the assessment template with embedded field notes.

Task	NPLF	Notes
PM.1.1	**P**	In some projects a vision document is provided and further discussed together with the stakeholders until a consensus is achieved. Official SOWs are not reviewed. Process/task not defined/documented. Tools: MS Word, email

Table 2 above present an example of the self-assessment template with field notes embedded into the Notes-column. The field notes do not segregate or identify how a particular piece of information was gained.

The data was mainly recorded in 2014. Hence, the study does not necessarily represent the actual current state of the case company's software processes. However, the study itself is valuable and can easily be reproduced with more current data if needed.

3 Quality

The purpose of this section is to explain that what is meant by quality in general and why quality is such an important concept. Furthermore, the evolution of quality from the early days to how modern quality is understood today, is briefly introduced.

3.1 What Is Quality

Quality has several interpretations and definitions, in fact there is no single industry wide accepted definition for quality. According to a book by Dale et al. (Dale, et al., 2013) many people think that they know what quality is, claiming that they can recognize a quality product when they encounter one. Thus, suggesting that quality can be

sensed. However, in reality quality is a far more complex concept, one that is quite difficult to comprehend. (Dale, et al., 2013.)

To truly understand quality, it helps to further discuss different views or definitions of quality. According to a book by Bath (Bath, 2010) quality is viewed as traditional and/or modern quality. Further, in (Dale, et al., 2013) multiple definitions of quality are discussed, such as qualitative and quantitative and fitness for use. In the following subsections some of the most important views of quality are discussed.

3.1.1 Qualitative Quality

Qualitative view of quality, while not necessarily that important, deserves its own section, since this is possibly one of the most common encounters of the term “quality” in everyday life. According to a book by Dale et al. (Dale, et al., 2013) the term “quality” is frequently used in a qualitative way for example in advertising and by people in general. However, such usage of the term “quality” can be considered highly subjective and even wrong.

3.1.2 Traditional Quality

Traditional and quantitative views of quality can be considered the same. Traditional quality focuses entirely on products and is controlled and measured according to an AQL (*Accepted Quality Level*) agreed between customer and its supplier. The idea is to inspect a batch of products according to an agreed sampling scheme and reject the entire batch if the number of defected products found in a sample exceeds the allowed amount. (Bath, 2010; Dale, et al., 2013.) Hence, according to a book by Dale et al. (Dale, et al., 2013) paradoxically traditional thinking defines quality by the number of defected products.

Traditional quality is perceived only as the quality of products, how well the products conform to specifications. It is thought of as features or properties of a product that can easily be inspected and measured. For example, does the diameter of a manufactured shaft fit within the specified tolerance. Traditional quality focuses on identifying the defected products through comprehensive system of inspections and removing them.

Methods for preventing defects from occurring in the first place are not considered. (Bath, 2010.)

However, the traditional thinking is inherently flawed. It suggests that quality could be achieved by inspecting, which is not possible, quality cannot be inspected into products. Contrary to traditional thinking, modern view of quality recognizes that to truly achieve quality, products need to be produced right the first time every time. (Bath, 2010.)

3.1.3 Modern Quality

Contrary to traditional reactive quality, modern quality is proactive. Modern quality concentrates on producing the products right at the first time. It focuses on the upstream processes, such as requirements and design, to prevent the delivery of defected products entirely. (Dale, et al., 2013.)

The modern approach to quality was originally heralded by a few modern thinkers, today known as the “Quality Gurus”. W. Edward Deming, Joseph M. Juran and Philip Crosby defined quality as “continuous improvement”, “fitness for use” and “conformance to requirements” respectively (Bath, 2010). Understanding and meeting, and hopefully exceeding, customer needs and their future requirements, and to continuously learn and to improve are in the core of modern quality thinking and TQM (Dale, et al., 2013).

In modern thinking quality is the responsibility of the entire organization, company management establishes and implements the QMS which is then applied by everyone in the organization. The organization needs to identify its key processes and their interactions. Furthermore, the processes are continuously improved based on objective performance metrics. (International Organization for Standardization, 2008a; Dale, et al., 2013.)

Modern quality management, as promoted by TQM and ISO 9001, is based on the following 8 QMPs (*Quality Management Principles*), adapted from (Dale, et al., 2013; Robitaille, 2010):

1. **Customer Focus** – TQM puts a great deal of emphasis on customer satisfaction. To improve customer satisfaction by successfully applying the QMS is a central concept of TQM.
2. **Leadership** – TQM states that it's the managements responsibility to establish and implement the QMS. The responsibility does not end after the initial decision. It means that management needs to be continuously involved in the application and improvement of the QMS.
3. **Involvement of People** – This QMP calls for the full engagement of people. Management is responsible for ensuring that, to work efficiently and fully engage, people have adequate training and resources.
4. **Process Approach** – Is defined by ISO as "An activity or a set of activities using resources, and managed in order to enable transformation of inputs into outputs, can be considered as a process" (International Organization for Standardization, 2008a). An organization has multiple processes that together form a system that produces a required outcome. These processes and their interactions need to be identified and they need to be managed together.
5. **Systems Approach to Management** – This QMP is strongly related to the previous QMP (Process Approach). In addition to core processes there are supporting processes to manage for example training needs, for monitoring the key process indicators, for internal audits etc.
6. **Continual Improvement** – The QMS needs to continually adapt to changing environment, to maintain Status Quo is not accepted. It is recognized that things change and that the company's processes need to adapt.
7. **Factual Approach to Decision Making** – With QMS decisions are based on facts (collected data, process metrics and audits) and not on subjective opinions. Thus, ensuring effective use of resources.
8. **Mutually Beneficial Supplier Relations** – This QMP stresses the importance of how a company interacts with and encompasses its suppliers. The way an organization interacts with its supplies is extremely important for the organization and its customers.

The model of a modern process based QMS is presented in Figure 2 below.

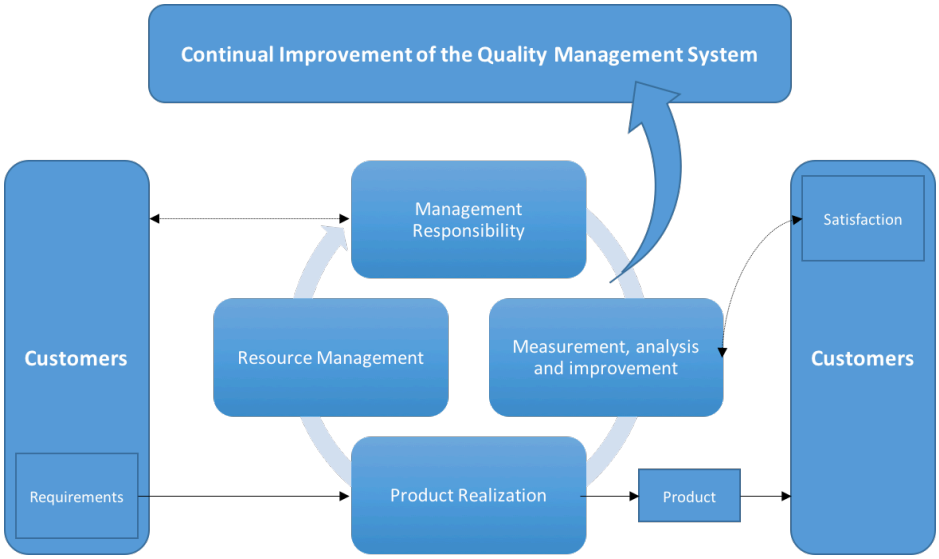

Figure 2. High level model of modern QMS. Adapted from (International Organization for Standardization, 2008a).

Figure 2 illustrates how the required processes interact and how understanding of customer requirements and customer satisfaction are central concepts in modern quality thinking. Furthermore, it illustrates the basic philosophy behind continuous improvement, the PDCA (*Plan-Do-Check-Act*) cycle or methodology (International Organization for Standardization, 2008a).

3.2 Why Is Quality Important

In the present economic and political climate customers are becoming increasingly demanding. The competition is not focused only on price but also on quality. (Bath, 2010.) Actually, it is seen that customers are willing to pay more for what they see as a quality product (Dale, et al., 2013). Furthermore, according to a standard by ISO (International Organization for Standardization, 2011a) quality standards are seen more and more as a contractual requirement.

All organizations are dependent on their customers. Thus, it can be said that customers are the most important reason for an organisation to exist. According to a book by Dale et al. (Dale, et al., 2013) a customer that is lost due to problems in quality is much harder to regain then a customer lost on a price based competition. Consequently, customers disappointed on quality can be lost for ever. (Dale, et al., 2013.)

PIMS (*Profit Impact of Market Strategy*) data is used to assess how a planned improvement translates to into profits. RPQ (*Relative Perceived Quality*) is a key PIMS concept which tells us how customers perceive a given product. (Dale, et al., 2013.) Studies show that RPQ has the greatest impact on ROI (*Return of Investment*), and further, that RPQ correlates with the relative market share. Thus, showing that companies with large market share have high quality and companies with low market share tend to have lower quality. (Horch, 2003.)

In addition to all of the above multiple studies show that quality improves business performance (Dale, et al., 2013). For example, a study carried out at the University of Bradford Management Centre (Letze, et al., 1997) showed that from 29 UK companies which display TQM characteristics 81 percent have an above median turnover (Dale, et al., 2013).

3.3 Evolution of Quality

The beginning of modern quality management can be traced back to 1924 when W. Shewhart, while working in Bell Laboratories, introduced statistical control charts and developed the concepts of process improvement. However, World War II was the real turning point that caused a rapid rise of quality awareness. (Bath, 2010; Kenett & Baker, 1999.)

After World War II in the 1950s W. Edwards Deming and Joseph M. Juran helped to rebuild the Japanese manufacturing industry. Deming introduced SQC (*Statistical Quality Control*) and Juran developed his quality trilogy. In the 1960s Philip Crosby and the concepts of “Zero-defects” and “do it right the first time” gained popularity. (Bath, 2010; Kenett & Baker, 1999.) Deming, Juran and Crosby are today considered as the “Quality Gurus” and top contributors to modern quality and TQM.

In the 1970s there was a dramatic shift from reactive inspection based quality assurance towards more strategic and proactive approach of removing the defects entirely (Bath, 2010). Thus, marking the beginning of the transition from traditional thinking to modern thinking.

Today the most prevalent approach to quality is TQM. As discussed earlier TQM focusses on customers and continuous improvement. Today TQM is implemented by the majority of professionally run manufacturing companies. (Kenett & Baker, 1999.)

3.4 Summary and Conclusions

Customers rule. An organization cannot exist without satisfying its customers. Thus, it is clear that to keep up with the ever changing, and increasingly demanding, customer requirements and expectations an organization is forced to introduce modern quality management principles into its processes. Companies reluctant in doing so will, sooner or later, lose their competitive edge.

This section discussed the evolution of quality and displayed some evidence on different quality approaches. However, the lessons learned are more or less from large manufacturing organizations. Since this study is about SMEs the next section is dedicated for discussing modern quality management in the context of SMEs.

4 SMEs and Quality

The purpose of this section is to discuss some of the special characteristics of SMEs and their effect on quality and quality management and how they should be taken into account with respect to quality.

4.1 What Is an SME

According to (Robitaille, 2010) there is no exact definition for SME. Nonetheless, EC (*European Commission*) categorizes SMEs into three distinct categories: Micro, small and medium sized enterprises. According to SME Definition Guide by the EC (European Union, 2015):

- Micro-enterprise has fewer than 10 employees and turnover not exceeding EUR 2 million

- Small-enterprise has fewer that 50 employees and a turnover not exceeding EUR 10 million
- Medium-enterprise has fewer than 250 employees and a turnover not exceeding EUR 50 million.

However, this distinction mainly exists due to economic and financial reasons. It is designed to help organizations in defining themselves as SMEs, and thus, allowing them to access various EU (*European Union*) support programmes targeted specifically for SMEs. (European Union, 2015.)

According to a guide by European Commission (European Union, 2015) SMEs are the engine of the European economy. SMEs are the main force behind innovation and entrepreneurial spirit. Moreover, around 90% of all enterprises are SMEs generating 2/3 out of all jobs in the EU. (European Union, 2015.) Furthermore, according to the OECD (*Organization for Economic Co-operation and Development*) the percentages are even higher in many countries around the world, accounting for 95 – 99% of the business population depending on the country (International Organization for Standardization, 2011a).

It can be argued though, that the above distinctions and categorizations are somewhat irrelevant in the context of quality and quality management. For example, an organization might have a large workforce performing some routine tasks and only a few people managing the operations. Should this kind of company then be considered as micro, small or medium? The important remark about SMEs is that in general they tend to share common characteristics that differentiate them from their larger counterparts. (Robitaille, 2010.)

4.2 Quality and Special Characteristics of SMEs

As discussed in Section 3 a central concept of modern quality is customer focus. Therefore, to be able to produce quality products an organization needs to understand and hopefully exceed the expectations of their customers. According to a study by Yan and Zhang (Yan & Zhang, 2011) managers and employees of SMEs contact their customers quite often and the obtained information is used to improve the quality of their

offerings. Thus, presenting SMEs with a clear competitive advantage in this respect (Yan & Zhang, 2011).

Another advantage of SMEs is the character of the internal communications. Typically, management and employees are collocated in a single site. Thus, communications tend to be more face-to-face and agile in fashion (Robitaille, 2010). The communication between employees, regardless of the hierarchy and rank, tends to be more open and frequent. Furthermore, employees find it generally easier to work in teams to achieve a common goal. (Yan & Zhang, 2011.)

SMEs tend to be very flexible and agile in nature which enables prompt responses to ever changing and new customer requirements. Meetings are typically informal and agreed changes are executed or implemented rapidly with minimal bureaucracy and documentation (Robitaille, 2010). This flexibility and agility can be considered a competitive advantage of SMEs. However, agility should not be gained at the expense of proper processes and documentation.

All of the above can be considered as competitive advantages of SMEs when compared to their larger counterparts. However, SMEs also face many challenges when striving for quality, for example limited resources. It is typical for employees to have multiple roles and to perform multiple different tasks. As a consequence, SMEs do not typically have dedicated and trained quality assurance personnel let alone a quality department. (Robitaille, 2010.)

4.3 Summary and Conclusions

Quality management is not exclusively for large organizations. There are no requirements or restrictions for the size of an organization to successfully implement quality management. Neither, are there any specific quality management guidelines designed for SMEs, all guidelines are generic and can be applied to all kinds of organizations. (International Organization for Standardization, 2008a.)

All organizations and their operational environments are different. Thus, when striving for quality an SME needs to consider its unique characteristics when designing their QMS. A QMS needs to reflect the organization it is serving. (International Organization for Standardization, 2008a.)

This section discussed quality from an SMEs perspective and common characteristics of SMEs were discussed. In the next section quality is discussed in the context of software. What are some special characteristics of software in the context of quality and how software quality can be achieved.

5 Software Quality

The purpose of this section is to briefly discuss the special characteristics of software industry with respect to quality and quality management. Furthermore, some of the most important software quality frameworks and standards are introduced.

The concepts of modern quality and TQM have evolved from the needs of traditional manufacturing industry (Chemuturi, 2014), which raises the question of their adequacy in the context of software industry. Since manufacturing industry and software industry are fundamentally different:

1. **Software is not produced** – In manufacturing industry the emphasis is on the production of goods. In contrast, software is almost 100% design and the production part is merely automated copying of the installation packages to some distribution media, such as DVDs, flash drives, online servers etc. (Dybå, et al., 2004; Oskarsson, 1999.)
2. **Software is complex** – Software solutions are highly complex when compared to other “ordinary” appliances. According to a text by Oskarsson (Oskarsson, 1999) “Todays software products are the most complex items created by humanity, with one exception: Our civilisations”.
3. **Product are unique** – The purpose, usage and operational environment of each software product is unique. Thus, exercised methodologies may need to be specifically adjusted for each project. (Dybå, et al., 2004.)

Figure 3 below further illustrates the differences highlighted in points 1 and 2 above. In the illustration the size of a rectangle represents the associated effort and/or costs.

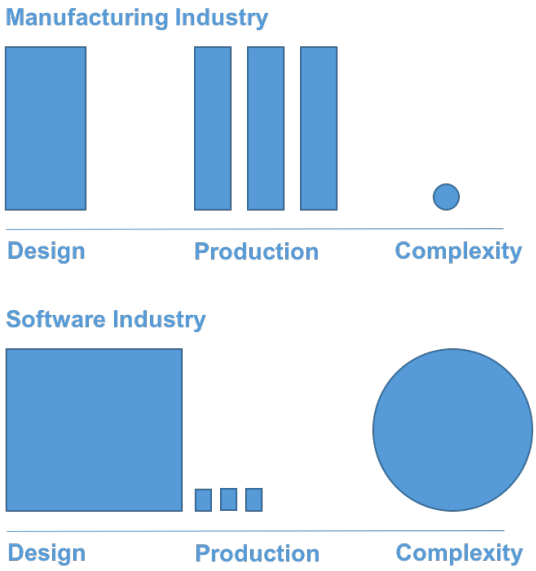

Figure 3. Differences between manufacturing and software industry. Adapted from (Oskarsson, 1999).

Figure 3 illustrates how in the manufacturing industry the production costs for a single item are relatively high in comparison to the design efforts. While in software industry the situation is completely the opposite. (Oskarsson, 1999.)

However fundamental the differences between manufacturing industry and software industry, the conclusion that can be drawn from the earlier sections in this study is that quality and quality management is not limited by the size of the organization nor the field it is operating on. All concepts of modern quality and TQM are also relevant in the context of software industry (Chemuturi, 2014).

5.1 Software Process Improvement

According to a book by Mutafelija and Stromberg (Mutafelija & Stromberg, 2003) in order for an SPI effort to be successful it needs to have measurable goals that are linked to respective business goals of the organization. Setting the goal to merely as achieving ISO 9001:2008 certification is not recommended. However, it can be considered as a suitable initial goal. (Mutafelija & Stromberg, 2003.)

There are multiple approaches for process improvement. Common for all SPI approaches is that they are organized in steps that give guidance on systematic process improvement, measurement of results and for continuous improvement. A typical SPI

approach, and the one that ISO 15504 is based on, has the following phases: (Mutafelija & Stromberg, 2003.)

1. Establishment of SPI goals and linking them to organizations business goals
2. Creation of a phased SPI project plan
3. Performing an assessment on the baseline processes
4. Analysing the assessment results and deriving a detailed SPI plan
5. Implementing the SPI plan
6. Measuring and monitoring the progress
7. Sustaining improvement gains
8. Monitor performance and continually improve.

In addition to SPI approaches there are multiple SPI frameworks to further guide the organization in achieving its process improvement goals. Frameworks bring detail to each SPI phase by introducing the best practises to be included in each phase. The frameworks and SPI approaches together provide a structured road-map for organizations process improvement efforts. (Mutafelija & Stromberg, 2003.)

5.2 Software Process Improvement Frameworks

According to a book by Kenett and Baker (Kenett & Baker, 1999) one of the major tasks of a SPI project is to identify the key areas for improvement. Without any structure or guidance of proven industry best practices this would be a chaotic endeavour. Consequently, various SPI frameworks and models to bring structure into the process have been developed. (Kenett & Baker, 1999.) In this section some of the most famous SPI frameworks are briefly introduced and compared.

5.2.1 ISO 9001

ISO 9001:2008 is an international QMS standard developed by ISO. It provides the requirements for a QMS that can be used for developing an organisations processes, for QMS certification and for contractual purposes. (International Organization for Standardization, 2008a.) ISO 9001 is the most prevalent QMS standard in the world, with over one million certified companies globally (Robitaille, 2010).

ISO 9001 was originally adapted from BS 5750, which was the first quality management standard in the UK. BS 5750 had been developed by the UK's Ministry of Defence and its purpose was to standardise how manufacturing processes are managed. The first version of ISO 9001 was released in 1987, named ISO 9001:1987 respectively. (The British Standards Institution, no date.)

The current revisions of ISO 9001 are general in nature, that is, the usage is not limited to any specific types or sizes of organisations nor to any specific products or businesses. (International Organization for Standardization, 2008a.) ISO 9001:2008 does not urge for all QMSs to be equal. Rather, ISO 9001:2008 stresses that all organizations and their operational environments are unique and that the QMS must reflect those unique characteristics. (Robitaille, 2010.) Thus, the application of ISO 9001:2008 is equally advantageous in software organizations as in manufacturing.

ISO 9001:2008 comprises five major requirement clauses 4 – 8, which each in turn contain several sub-clauses. These clauses together form the requirements for ISO 9001:2008. To achieve ISO 9001:2008 compliance an organization needs to fulfil all the requirements. However, exclusions can be considered, but are limited to requirements within Clause 7. (International Organization for Standardization, 2008a.)

5.2.2 CMMI

SEI (*Software Engineering Institute*) at Carnegie Mellon University has developed a number of CMMI (*Capability Maturity Model Integration*) models for different purposes. Such as:

- **CMMI-DEV** – CMMI for development
- **CMMI-SVC** – CMMI for services
- **CMMI-ACQ** – CMMI for acquisition

CMMI has evolved from the original CMM for software. CMM for software originated from the needs of the U.S. Department of Defence when it recognized the need to be able to more precisely predict the performance of its software providers. The first version of CMM for software was released in 1991, named CMM for software version 1.0. (Mutafelija & Stromberg, 2003.)

Essentially CMMI models are collections of industry and government best practises aimed at helping an organization to improve its processes. The CMMI model of interest in the context of software is CMMI-DEV. CMMI-DEV focuses on best practises for developing quality products and services that meet or exceed the customer expectations. CMMI-DEV contains the best practises and integral elements of effective processes and presents an evolutionary path for improvement from chaotic to mature processes. (Software Engineering Institute, Carnegie Mellon, 2010.)

CMMI-DEV model comprises 22 process areas and maturity levels 1 – 5. Each maturity level contains the process areas required for an organization to be appraised for that given maturity. Furthermore, CMMI-DEV implements a staged and a continuous model for process improvement. (Foegen & Richter, 2003.)

5.2.3 ISO 15504 / SPICE

ISO 15504 / SPICE (*Software Process Improvement and Capability dEtermination*) is an international standard on software process assessment developed by ISO. It has originated from the need to harmonize various software process assessment models such as CMMI and Bootstrap (Mutafelija & Stromberg, 2003). The first draft of ISO 15504 was released as technical report in 1998 (Foegen & Richter, 2003).

ISO 15504 is essentially a process assessment standard that allows the usage of other compatible assessment models. It is harmonized with CMMI, hence, either one can be used for process assessment (Foegen & Richter, 2003). ISO 15504 contains a reference model and a guide on using assessments for software process improvement and capability determination. ISO 15504 is compatible with ISO 9001 and ISO 12207 and has served as a basis in development of many assessment models, such as CMMI (Mutafelija & Stromberg, 2003).

In the core of ISO 15504 is the two dimensional reference model that is used to assess individual processes and their capabilities. The reference model comprises of 40 processes that are divided into 5 categories that are further assessed for a capability level between 0 – 5 that is based on 9 process attributes. ISO 15504 implements a continuous model for process improvement. (Foegen & Richter, 2003.)

5.2.4 ISO 12207

ISO 12207 “Information Technology - Software Lifecycle Processes” is an international standard developed by ISO. Its purpose is to provide a common framework and vocabulary for software lifecycle processes that can be referenced by the software industry. The first version of the standard ISO 12207:1995 was published in 1995. (International Organization for Standardization, 2008b.)

The current version ISO 12207:2008 provides a comprehensive set of lifecycle processes, process purposes, activities, tasks and process outcomes. Furthermore, it provides a PRM (*Process Reference Model*) that can be used in accordance with ISO 15504 when performing process assessments. (International Organization for Standardization, 2008b.)

5.2.5 ISO 29110

ISO 29110 “Software Engineering – Lifecycle Profiles for Very Small Entities (VSEs)” is a series of international standards and guides developed by ISO. The first version of ISO 29110 was published in 2011. (International Organization for Standardization, 2011a.)

According to ISO 29110 “VSE is an entity (enterprise, organization, department or project) having up to 25 people“ (International Organization for Standardization, 2011a).

In studies conducted studies it has been found that international standards do not fit the needs of SMEs/VSEs. Most of the VSEs cannot afford the required resources. Consequently, it is almost impossible for a VSE to achieve conformance with international standards. Thus, due to the lack of conformity with standards, VSEs are not able to participate in project where conformance is required. (International Organization for Standardization, 2011a.)

The purpose of ISO 29110 is to provide guidance for VSEs and to alleviate some of the challenges that VSEs are facing. ISO 29110 VSE profiles are essentially subsets of existing international standards (such as ISO 12207) that are relevant in the context of VSEs. (International Organization for Standardization, 2011a.)

5.2.6 Frameworks - Summary

A number of SPI frameworks exists to address various needs of software suppliers and their customers. Some more general and some more prescriptive. However, there are some essential commonalities. Firstly, they are based on the concepts of modern quality and TQM, focusing on customers and continuous process improvement. Secondly, they provided guidance and best practises for process improvement, not concrete processes. Some of the most important attributes of the given frameworks are compared in Table 3 below.

Table 3. Comparison of SPI frameworks.

	ISO 9001	**CMMI-DEV**	**ISO 15504**	**ISO 12207**	**ISO 29110**
Developer	ISO	SEI	ISO	ISO	ISO
Year published	1987	As CMM for software in 1991	As a technical report in 1998	1995	2011
Current version	ISO 9001:2015	CMMI-DEV, V1.3	EOL, replaced by ISO 33001:2015	ISO 12207:2008	ISO 29110-1:2011
Concept	International QMS Standard	Process improvement model	International process assessment standard	Common framework for software lifecycle processes	Lifecycle profiles for VSEs
Scope	General	Software	Software	Software	Software
Implementation model	Flexible, full conformance required	Focused, staged and continuous improvement models	Flexible, continuous improvement model	-	-
Focus	Customers and processes	Business and processes	Processes	Processes	Processes
Compatibility	Conforms to CMMI level 3	Compatible with ISO 15504	Compatible with CMMI and ISO 9001	Compatible with ISO 15504	Compatible with ISO 12207

	ISO 9001	CMMI-DEV	ISO 15504	ISO 12207	ISO 29110
Prevalence	Mostly Europe	Mostly U.S.	Mostly Europe	Mostly Europe	Globally

5.3 Summary and Conclusions

In this section quality was discussed in the context of software. Some of the most essential differences between manufacturing and software industry, from quality perspective, were highlighted. Furthermore, some of the most important SPI frameworks were introduced and compared.

To reiterate, this study focuses on SPI based on ISO 9001:2008 in small a solution provider company operating in Espoo Finland that has recently been awarded an ISO 9001:2008 certificate. As introduced, ISO 9001:2008 is a general and flexible SPI framework, and further, it is the most prevalent SPI framework in Europe. Hence, ISO 9001:2008 can be considered a suitable SPI framework for the case company. In the following section ISO 9001:2008 and its applicability to software is further discussed.

6 ISO 9001:2008 and Software

In this section the ISO 9001:2008 QMS standard is discussed in more detail, mainly from a software perspective. The general introduction, already given in Section 5.2.1, is not reiterated here. Rather, first the evolution of ISO 9001:2008 is discussed from a software perspective, and second, ISO 9001:2008 requirements and their application to software are discussed.

6.1 Evolution of ISO 9001 from Software Perspective

The first version of ISO 9001 was released in 1987 named ISO 9001:1987. The standard was originally designed for the needs of manufacturing industry. (Oskarsson, 1999.) Since manufacturing industry and software industry are very different in nature, ISO 9001:1987 did not reflect the software lifecycle very well (Suryn, et al., 2004). The emphasis in manufacturing industry is on the production activities while, when building

software, the emphasis is almost completely on design. Thus, in the early days the application of ISO 9001 to software development and maintenance was found to be problematic. (Oskarsson, 1999.)

The challenges for software industry to adopt ISO 9001:1987 standard were noted quite early on. To be able to build quality software the involved processes needed to be identified and a special interpretation for applying ISO 9001 to software development and maintenance was needed. (Oskarsson, 1999.) Consequently, in 1997 the first version of ISO 9000-3 *"Quality management and quality assurance standards - Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software (ISO 9000-3:1997)"* was published. (Suryn, et al., 2004.)

Mean while ISO 12207:1995 Software Lifecycle Processes standard had been developed and ISO 9001:1994, the second revision of ISO 9001, had been released. The guidance in ISO 9000-3:1997 was largely based on standardised lifecycle processes from ISO 12207:1995 and matched all ISO 9001:1994 requirements. As a result, it formed a usable framework for mapping ISO 9001:1994 requirements with suitable software processes of ISO 12207:1995. (Suryn, et al., 2004.)

Since 1990s' the software engineering standards have advanced tremendously. The joint technical committee ISO/IEC JTC 1, which is the committee responsible for information technology, has developed and released multiple standards and technical reports to support ISO 9001. Today ISO 9001:2015, the fifth revision of ISO 9001, has already been released. ISO 9000-3 has received its own ISO/IEC number 90003 and is now at its second revision ISO 90003:2014. ISO 12207 has experienced a few amendments and the second revision ISO 12207:2008 is currently out.

To avoid confusion, the subject of this study should be reiterated here: This study concentrates on the fourth revision ISO 9001:2008, even though the fifth revision of the standard (ISO 9001:2015) has already been released. This is due to the fact that the company, that this study is conducted for, has been certified to ISO 9001:2008. Hence, for the software processes to achieve compatibility with ISO 9001:2008 is the company's primary goal. Therefore, this study focuses on ISO 9001:2008 requirements.

6.2 ISO 9001:2008 Requirements

In this section the ISO 9001:2008 requirements are discussed. The intent is not to repeat all the details of the requirements, but rather to summarize the key points. Details for the requirements can be found in the standard (International Organization for Standardization, 2008a).

6.2.1 Introduction and Scope

Section 1 of ISO 9001:2008 standard (International Organization for Standardization, 2008a) specifies the scope of the standard and the requirements. The requirements specified in the standard concern organizations that:

1. Need to demonstrate their ability to consistently produce products that meet the customers and any applicable statutory and regulatory requirements.
2. Aim to continuously improve customer satisfaction through effective application of their QMS system and commitment to continuous improvement.

Since this study was conducted for an SME company, it is worth reiterating here what was already stated earlier in Section 5.2.1:

1. Section one of ISO 9001:2008 (International Organization for Standardization, 2008a) clearly states that the standard is meant to be general in nature, that is, its usage is not limited to any specific types or sizes of organisations nor to any specific products or businesses. Even companies consisting of a single person are known to be certified to ISO 9001:2008 (Robitaille, 2010).
2. In the introduction of ISO 9001:2008 (International Organization for Standardization, 2008a) it is noted that all organizations and their operational environments are unique and that the QMS must reflect those unique characteristics.

ISO 9001:2008 also allows for exclusion of some of its requirements, depending on the unique characteristics of the company. However, these exclusions are restricted to the requirements of Clause 7 (International Organization for Standardization, 2008a). Requirements in all other sections concern all organizations (Robitaille, 2010).

ISO 9001:2008 comprises of five major requirement clauses 4 – 8, which each in turn contain several sub-clauses:

- 4 Quality Management System
- 5 Management Responsibility
- 6 Resource Management
- 7 Product Realization
- 8 Measurement, Analysis and Improvement

These clauses together form the requirements for ISO 9001:2008. Each of the major clauses is introduced in more detail in the following sections.

6.2.2 Quality Management System

In Clause 4 of ISO 9001:2008 (International Organization for Standardization, 2008a) the general requirements for the QMS and requirements for the documentation are given. Clause 4 is organized into two sub-clauses:

- 4.1 General requirements
- 4.2 Documentation requirements

The requirements of ISO 9001:2008 Clause 4 are summarized in Table 4 below.

Table 4. ISO 9001:2008 requirements for quality management system. Adapted from (Whittington, no date).

4 Quality Management System
4.1 General requirements • The organization shall identify and implement all of its processes and determine their sequence and interactions. • The processes shall be monitored and analysed to achieve planned results and continual improvement. • The processes shall be managed according to ISO 9001:2008. • The organization shall also control all outsourced processes
4.2 Documentation requirements • **4.2.1 General** o Organizations quality policy and objectives and the quality manual shall be documented. o All required procedures and records shall be documented along with any substantive organization specific documents and records • **4.2.2 Quality manual** o A quality manual shall be established and maintained o It shall include the scope of the QMS with any exclusions, documented procedures and interactions between its processes • **4.2.3 Control of documents**

○ Required documents and records shall be controlled and procedures documented ○ Document approval, review, update, versioning, legibility, identifiability and distribution needs to be controlled and ensured • **4.2.4 Control of records** ○ Recorded evidence of conformity and effective operation of the QMS shall be controlled and procedures documented

Table 4 above summarises the requirements of ISO 9001:2008 Clause 4. It states that the organization shall implement and manage its QMS processes in accordance to ISO 9001:2008 requirements and that all required procedures shall be documented. Further, a quality manual shall be documented and all documents and records shall be controlled and evidence of the effective operation of the QMS shall be controlled and procedures documented. (International Organization for Standardization, 2008a.)

6.2.3 Management Responsibility

In Clause 5 of ISO 9001:2008 (International Organization for Standardization, 2008a) the requirements for different responsibilities of the top management are given. Requirements from management commitment to the QMS to QMS reviews by top management are defined. Clause 5 is organized into six sub-clauses:

- 5.1 Management commitment
- 5.2 Customer focus
- 5.3 Quality policy
- 5.4 Planning
- 5.5 Responsibility, authority and communication
- 5.6 Management review

The requirements of ISO 9001:2008 Clause 5 are summarized in Table 5 below.

Table 5. ISO 9001:2008 requirements for management responsibility. Adapted from (Whittington, no date).

5 Management Responsibility
5.1 Management commitment • Top management shall provide evidence of its commitment to developing the QMS • Communicate the importance of meeting requirements • Establish a quality policy and objectives • Conduct management reviews • Ensure resources **5.2 Customer focus** • Top management shall ensure that requirements are determined and met **5.3 Quality policy** • Top management shall ensure that the quality policy is fit for the organization and includes a commitment to comply to requirements and continuous improvement and is continuously re-

viewed
- Quality policy shall provide a framework for establishing and reviewing quality objectives and needs to be understood within the organization

5.4 Planning
- **5.4.1 Quality objectives**
  - Top management shall ensure that measurable quality objectives are established at relevant functions and levels
- **5.4.2 Quality management system planning**
  - Top management shall ensure that the QMS is planned according to the given requirements and objectives and that QMS integrity is maintained during changes

5.5 Responsibility, authority and communication
- **5.5.1 Responsibility and authority**
  - Top management shall ensure that authorities are defined and communicated
- **5.5.2 Management representative**
  - Top management shall appoint a member of the management who is responsible for QMS processes, reporting on QMS performance and improvements and promoting the awareness of customer requirements
- **5.5.3 Internal communication**
  - Top management shall ensure that communication processes are established and effectiveness of the QMS is communicated

5.6 Management review
- **5.6.1 General**
  - Top management shall periodically review the QMS for its suitability, performance and improvement
  - Records of the review shall be maintained
- **5.6.2 Review input**
  - Input to review shall include audit results, customer feedback, process performance and product conformity, status of preventive and corrective actions, follow-up action from previous reviews, changes to QMS and recommended improvements
- **5.6.3 Review output**
  - Output shall include decisions and actions related to QMS improvement, products and resources

Table 5 above summarises the requirements of ISO 9001:2008 Clause 5. It states that top management shall be committed to developing the QMS and establishment of the quality policy and provide evidence for its commitment. Top management shall ensure that the QMS and quality policy it fit for the organization and importance of different quality aspects and quality objectives are communicated and understood. Further, top management shall ensure framework for continuous improvement and that required authorities are defined and communicated. (International Organization for Standardization, 2008a.)

6.2.4 Resource Management

In Clause 6 of ISO 9001:2008 (International Organization for Standardization, 2008a) the requirements for managing different types of resources are given. The requirements from provisioning of resources to requirements for work environment are defined. Clause 6 is organized into four sub-clauses:

- 6.1 Provision of resources

- 6.2 Human resources
- 6.3. Infrastructure
- 6.4 Work environment

The requirements of ISO 9001:2008 Clause 6 are summarized in Table 6 below

Table 6. ISO 9001:2008 requirements for resource management. Adapted from (Whittington, no date).

6 Resource Management
6.1 Provision of resources • Resources to maintain and improve the QMS and to enhance customer satisfaction shall be determined and provided
6.2 Human resources • **6.2.1 General** o Personnel affecting the conformity to product requirements shall be competent • **6.2.2 Competence, training and awareness** o Organization shall determine the necessary competence and provide necessary training o Evaluate the effectiveness of actions taken and maintain records of education, training, skills and experience
6.3 Infrastructure • The organization shall maintain appropriate infrastructure needed to meet product requirements, such as buildings, workspace, utilities, hardware and software, and supporting services
6.4 Work environment • The organization shall maintain appropriate environment needed to meet product requirements

Table 6 above summarises the requirements of ISO 9001:2008 Clause 6. It states that competent and adequate resources shall be determined and provided and that any required training shall be provided. Further, appropriate infrastructure and work environment shall be maintained. (International Organization for Standardization, 2008a.)

6.2.5 Product Realization

In Clause 7 of ISO 9001:2008 (International Organization for Standardization, 2008a) the requirements for different aspect of product realization are given. Requirements from planning of product realization processes to controlling the measuring equipment are defined. Clause 7 is organized into six sub-clauses:

- 7.1 Planning of product realization
- 7.2 Customer-related processes
- 7.3 Design and development
- 7.4 Purchasing
- 7.5 Production and service provision
- 7.6 Control of monitoring and measuring equipment

The requirements of ISO 9001:2008 Clause 7 are summarized in Table 7 below.

Table 7. ISO 9001:2008 requirements for product realization. Adapted from (Whittington, no date).

7 Product Realization

7.1 Planning of product realization

- The organization shall develop processes for product realization that are consistent with the QMS
- Quality objectives, requirements and resources specific to the product shall be determined
- Monitoring, inspection, tests and acceptance criteria and records needed to provide evidence shall be determined

7.2 Customer-related processes

- **7.2.1 Determination of requirements related to the products**
  - The organization shall determine customer requirements specified and unspecified customer requirements and applicable statutory and regulatory requirements
- **7.2.2 Review of requirements related to product**
  - Product requirements shall be reviewed to ensure they are defined and can be met
  - Records of the review results and corresponding actions shall be maintained
  - Customer requirements shall be confirmed and any changes shall be communicated to relevant personnel
- **7.2.3 Customer communication**
  - Effective communications with the customer concerning product information, enquiries, contracts, orders and feedback shall be ensured

7.3 Design and development

- **7.3.1 Design and development planning**
  - The organization shall plan and control the design and development of product
  - Design and development stages and corresponding reviews, verifications and validations and responsibilities shall be determined
- **7.3.2 Design and development inputs**
  - Inputs relating to product requirements shall be determined and records maintained including functional, performance, statutory, regulatory and any other requirements
  - Requirements shall be complete, unambiguous and not in conflict with each other
- **7.3.3 Design and development outputs**
  - The design and development outputs shall be suitable for verification against the inputs and shall be approved
  - Acceptance criteria shall be referenced and safe and proper use of the product shall be characterized
- **7.3.4 Design and development review**
  - Systematic reviews of design and development shall be performed **t**o evaluate that requirements are met and to identify problems
- **7.3.5 Design and development verification**
  - Verification shall be performed in accordance with planned arrangements and records of results shall be maintained
- **7.3.6 Design and development validation**
  - Design and development validations shall be performed in accordance with planned arrangements prior to implementation and records of the results shall be maintained
- **7.3.7 Control of design and development changes**
  - Design and development changes shall be identified and records maintained. Changes shall be reviewed and approved prior to implementation and records of the reviews shall be maintained

7.4 Purchasing

- **7.4.1 Purchasing process**
  - The organization shall ensure that purchased product conforms to purchase requirements
  - Suppliers shall be evaluated according to established requirements and criteria and records of evaluations shall be maintained
- **7.4.2 Purchasing information**
  - Purchasing information shall describe the product to be purchased
  - Requirements for approval, qualification of personnel and QMS shall be included in the purchasing information
- **7.4.3 Verification of purchased**

- The organization shall establish inspections to ensure that purchased product meets requirements

7.5 Production and service provision

- **7.5.1 Control of production and service provision**
  - Production and service provisioning shall be carried out under controlled conditions
  - Controlled conditions shall include the availability of information, instructions, equipment, monitoring and measurement and release, delivery and post-delivery activities
- **7.5.2 Validation of processes for production and service provision**
  - Processes shall be validated where the resulting output cannot be verified and deficiencies become apparent only after delivery
  - Arrangements for these processes shall include criteria for review and approval, approved equipment and qualification of personnel, specific methods and procedures and requirements for records
- **7.5.3 Identification and traceability**
  - The product and its status shall be identifiable throughout product realization
- **7.5.4 Customer property**
  - Care shall be exercised with customer property while it is under the organizations control
- **7.5.5 Preservation of product**
  - Product and its constituent parts shall be preserved during internal processing and delivery

7.6 Control of monitoring and measuring equipment

- Monitoring and measuring to be undertaken and the equipment needed shall be determined
- Processes for monitoring and measurement shall be established
- To ensure valid results the equipment shall be calibrated, re-adjusted and protected damage and detoriation
- Measuring result shall be assessed for validity and appropriate action shall be taken
- Records of calibration and verifications shall be mainteined

Table 7 above summarises the requirements of ISO 9001:2008 Clause 7. It states that processes for product realization shall be determined and that quality objectives, resources and requirements for the product shall be determined and the means to provide evidence shall be determined. Further, it states that customer requirements shall be determined and reviewed and effectively communicated and appropriate records maintained. The design and development of the product shall be carried out in a controlled and planned manner with appropriate reviews, verifications and validations at different stages and proper change control in place. (International Organization for Standardization, 2008a.)

The production shall take place under controlled conditions with all required information, equipment etc. readily available. The production processes shall be validated including methods, equipment, qualifications and requirements for records. Furthermore, the product and its status shall be traceable. There is also a number of requirements for controlling the purchasing process and suppliers and of controlling the monitoring and measurement equipment. (International Organization for Standardization, 2008a.)

6.2.6 Measurement, Analysis and Improvement

In Clause 8 of ISO 9001:2008 (International Organization for Standardization, 2008a) the requirements for different aspects of monitoring, measurement and continuous improvement are given. Requirements from monitoring and measuring customer satisfaction, processes and products to analysis and improvement are defined. Clause 8 is organized into five sub-clauses:

- 8.1 General
- 8.2 Monitoring and measurement
- 8.3 Control of nonconforming products
- 8.4 Analysis of data
- 8.5 Improvement

The requirements of ISO 9001:2008 Clause 8 are summarized in Table 8 below.

Table 8. ISO 9001:2008 requirements for measurement, analysis and improvement. Adapted from (Whittington, no date).

8 Measurement, analysis and improvement
8.1 General • The organization shall implement the processes needed to demonstrate conformity to product requirements, ensure conformity of the QMS and to continuously improve the QMS **8.2 Monitoring and measurement** • **8.2.1 Customer satisfaction** o Methods for obtaining information on customer perception as to whether the product meets the requirements shall be determined • **8.2.2 Internal audit** o The organization shall conduct internal QMS audits at planned intervals o Audit programme shall be planned. The audit criteria, scope, frequency and methods shall be defined o A documented procedure shall be established defining the responsibilities and requirements for audits o Records of the audits shall be maintained o Management shall ensure that any corrective actions are taken without undue delay • **8.2.3 Monitoring and measurement of processes** o Suitable methods for QMS monitoring and measurement shall be applied • **8.2.4 Monitoring and measurement of product** o The organization shall monitor and measure the characteristics of the product to ensure that product requirements have been met. Records shall be maintained indicating the persons authorizing the delivery o The delivery shall not proceed until the planned arrangements have been completed **8.3 Control of nonconforming product** • The organization shall ensure that nonconforming products are identified and not delivered unintentionally and a documented procedure for dealing with nonconforming products shall be established • Nonconforming products shall be dealt with some of the following ways: eliminate the nonconformity, authorizing its use, preclude its original intended use and by taking appropriate actions if non-conformance is detected after delivery • Records of nonconformities and actions taken shall be maintained **8.4 Analysis of data**

- Appropriate data to demonstrate the effectiveness of the QMS shall be collected and analysed and evaluated for continuous improvement

8.5 Improvement

1. **8.5.1 Continual improvement**
   - The organizations shall continuously improve the QMS
2. **8.5.2 Corrective action**
   - Actions shall be taken to eliminate the causes of nonconformities
   - Documented procedures shall be established to define requirements for reviewing nonconformities, determining the causes, evaluating the need for actions to prevent reoccurrence and records of actions taken and reviewing the effectiveness
3. **8.5.3 Preventive action**
   - Actions to eliminate the causes of potential nonconformities and their re-occurrence shall be determined
   - Documented procedures shall be established to define requirements for determining potential nonconformities and their causes, actions needed to prevent their occurrence and records of results of actions taken and reviewing the effectiveness

Table 8 above summarises the requirements of ISO 9001:2008 Clause 8. It states that the organization shall implement the processes needed to demonstrate conformity to product requirements and to ensure QMS conformity and continuous improvement. Customer satisfaction, processes and products shall be monitored and measured and internal audits shall be performed and continuously improved. Furthermore, nonconforming products shall be controlled to prevent their unintentional delivery. (International Organization for Standardization, 2008a.)

6.3 ISO 9001:2008 Application to Computer Software

As discussed in earlier sections of this study ISO 9001:2008 is a general QMS standard that has its roots in manufacturing industry. Hence, its application to software development and maintenance requires further guidance and interpretation. Thus, ISO 90003:2014 “Software engineering – Guidelines for the application of ISO 9001:2008 to computer software” has been developed.

As its name suggests, ISO 90003 is not an actual standard, but a guide. It provides guidance for the application of ISO 9001:2008 to acquisition, supply, development, operation and maintenance of computer software. Further, it identifies the issues that should be addressed when applying ISO 9001 in the context of software. The guidance given in ISO 90003 is general and non prescriptive independent of any specific technologies, lifecycle models, development processes etc. (International Organization for Standardization, 2014.)

ISO 90003:2008 relies largely on ISO 12207:2008 and in each clause of ISO 90003:2008 the respective processes and process outcomes of 12207:2008 are refer-

enced. Hence, it provides a mapping between ISO 9001:2008 requirements and corresponding software lifecycle processes and process outcomes from ISO 12207:2008. Thus, enabling the organizations applying ISO 9001:2008 in the context of software to use processes from ISO 12207:2008 to support the ISO 9001:2008 process model. (International Organization for Standardization, 2014.)

The SPI approach discussed earlier in Section 5.1 would typically be applied when applying ISO 9001:2008 in the context of software. After the goals and plans are in place, a process assessment is performed for the current process baseline. ISO 15504, introduced in Section 5.2.3, could be used in the assessment. ISO 15504 contains an exemplar PAM (*Process Assessment Model*) that is founded on industry best practises and that further references and supports ISO 12207 software lifecycle processes. (Suryn, et al., 2004.) Figure 4 below shows the relationships between ISO 9001 and ISO software engineering standards.

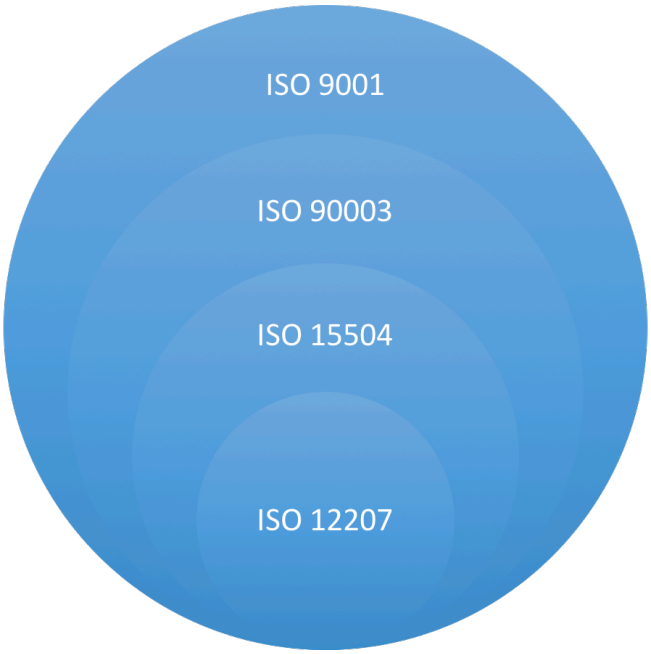

Figure 4. The relationships between ISO standards.

Figure 4 shows how ISO 9001 provides the general requirements for the QMS and ISO 90003 further provides the required guidance and interpretation for the application of ISO 9001 in the context of software. Furthermore, ISO 15504 provides framework for process assessment and best practises to support the software lifecycle processes provided by ISO 12207. (Suryn, et al., 2004.)

6.4 Summary

In this section the ISO 9001:2008 QMS standard was discussed in more detail. Its evolution from the software perspective was discussed and the requirements were introduced. Moreover, the application of ISO 9001:2008 to software and some of the supporting ISO software engineering standards were discussed in general. In the following sections ISO 9001:2008 based SPI is discussed in the context of the case company.

7 Setting the Scope for SPI

A complete SPI project contains several phases that are outside of the scope of this study. The purpose of this section is to highlight the SPI scope of this study and to elicit that only a subset of the typical phases in a SPI project are addressed.

7.1 Defining Relevant Phases

To reiterate, the purpose of this study is to find the key software process areas for improvement and to provide recommendations for SPI. Hence, most of the process improvement phases that are normally involved in an SPI project can be completely or partially excluded (see Section 5.1 of this study for a typical SPI approach). The phases that are completely excluded from the scope are:

- **2. Creation of a phased SPI project plan** – Only recommended SPI actions are provided. Planning and scheduling of the actual implementation of the recommended SPI actions is outside of the scope of this study.
- **5. Implementing the SPI plan** – Only recommended SPI actions are provided. Implementation of the recommended SPI actions is outside of the scope of this study.
- **6. Measuring and monitoring the progress** – Since implementing the recommended SPI actions is outside of the scope of this study, so are measuring and monitoring of the progress.
- **7. Sustaining improvement gains** – Since implementing the recommended SPI actions is outside of the scope of this study, so is sustaining the improvement gains.

- **8. Monitor performance and continually improve** - Since implementing the recommended SPI actions is outside of the scope of this study, so are monitoring and continuous improvement.

And the phases that are fully or partially included are:

- **1. Establishment of SPI goals and linking them to organizations business goals** – The goals need to be established in order to have a reference in the assessment.
- **3. Performing an assessment on the baseline processes** – Current processes need to be assessed to identify the shortcomings when compared to the established goal.
- **4. Analysing the assessment results and deriving a detailed SPI plan** – The assessment results need to be analysed to be able to recommend SPI actions.

The SPI phases that are included in the scope of this study were identified above. In the following sections the goal for SPI is established.

7.2 Establishing the SPI Goal

As discussed in Section 5.1, SPI goals should be based on the business goals of the organization. However, defining business goals for the case company and linking them to the SPI goals is outside of the scope of this study. Setting the goal to merely as achieving ISO 9001:2008 certification is not recommended. However, given the purpose of this study it can be considered as a suitable goal.

As discussed in Section 5.2.1, for an organization to achieve ISO 9001:2008 certification full conformance with the standard is required. Exclusions can be considered only on some parts of Clause 7. (International Organization for Standardization, 2008a.) Hence, the goal for SPI in this study can be established as: Identify key SPI areas and required SPI actions to take for the case company to achieve full conformance with ISO 9001:2008 requirements.

7.3 Summary

In this sections the SPI phases that are included in the scope of this study were identified and the goal for SPI was established. In the following sections an assessment is performed and recommendations for SPI are derived.

8 Current State of Software Process and SPI Recommendations

In this section the key SPI areas are identified and respective SPI recommendations are provided. First a method for performing the self-assessment is selected and a model for identifying the key SPI areas is introduced. Second, the self-assessment is performed by applying the model, the results are analysed and the key SPI areas are identified. Third, a suitable model for performing the gap analysis and a method for deriving the SPI recommendations are discussed and introduced. Finally, the gap analysis is performed by applying the model and the SPI recommendations a derived and presented.

8.1 Selecting Self-Assessment Method

Multiple methods for performing a software process assessment exists. In the following sub-sections some of the methods are discussed, and finally, a suitable method for the purposes of this study is selected.

8.1.1 Method Based on ISO 15504

As discussed in Section 5.1 and Section 6.3, a reliable and repeatable way for conducting a process assessment would be the application of a suitable assessment framework. ISO 15504 introduced in Section 5.2.3 is such a framework. It provides the best practises that support ISO 12207 software lifecycle processes, and further, incorporates an exemplar PAM that could be used in the assessment. Furthermore, it is compatible with ISO 9001.

However, a major disadvantage of ISO 15504 in the context of this study is that it is not publicly and/or freely available. Furthermore, as discussed in Section 5.2.3, the PAM incorporated in ISO 15504 is generic in nature and is based on the best practices of the industry. Given the previous and the origins of ISO 15504, it can be argued that ISO 15504 and the exemplar PAM are probably not ideal for assessing the processes of an SME.

8.1.2 Method Based on ISO 29110 and a PAM for VSEs

According to a study by Völcker et al. (Völcker, et al., 2002) better results may be gained, if a domain specific customized PAM is used in the process assessment. However, developing a customized PAM is outside of the scope of this study.

The approach introduced in a study by Varkoi and Mäkinen (Varkoi & Mäkinen, 2010) "A Process Assessment Model for Very Small Software Entities", where an attempt is made in developing a PAM for VSEs, seems to be more suitable. However, it seems that the goal in the study (Varkoi & Mäkinen, 2010) is to develop a PAM that is suitable for formal assessments and that the exemplar VSE PAM introduced still has several limitations.

8.1.3 Method Based on ISO 29110 Basic VSE Profile

According to the DP (*Deployment Package*) (Varkoi, 2009), a formal assessment is required when the organizations process capabilities need to determined in an objective and repeatable manner. On the contrary, a self-assessment is suitable when an organization needs to study its processes as the basis for process improvement, which is the case in this study.

The DP (Varkoi, 2009) contains a method for performing a self-assessment that is based on ISO 29110 Basic VSE profile (International Organization for Standardization, 2011b). The profile defines two processes, PM (*Project Management*) and SI (*Software Implementation*). Each of the processes have objectives that are achieved by carrying out the tasks associated with the process activities. In the self-assessment it is these tasks that are assessed and rated. (International Organization for Standardization, 2011b.)

The method introduced has the following characteristics that fit the purposes of the present study:

- It is based on ISO 29110 and hence takes into account the small size of an organization, which is the main defining characteristic of the case company (see Section 5.2.5 for introduction to ISO 29110 and VSEs).
- It is a suitable method for performing a self-assessment for SPI purposes.

Thus, the self-assessment method introduced in the DP (Varkoi, 2009) is selected as the method for performing the self-assessment.

8.2 Developing Model for Identifying Key SPI Areas

As discussed in Section 5.2.5, the ISO 29110 VSE profiles are essentially subsets of existing international standards (such as ISO 12207) that are relevant in the context of VSEs. Furthermore, as discussed in Section 6.3, ISO 90003 also relies largely on ISO 12007. Hence, it can be argued that ISO 29110 Basic VSE Profile and ISO 90003 (and ISO 9001) overlap, and it is this overlapping that essentially forms the key process areas of a SME/VSE with respect to ISO 9001 requirements. The model for finding the key SPI areas of a SME/VSE based on ISO 9001:2008 requirements is presented in Figure 5 below.

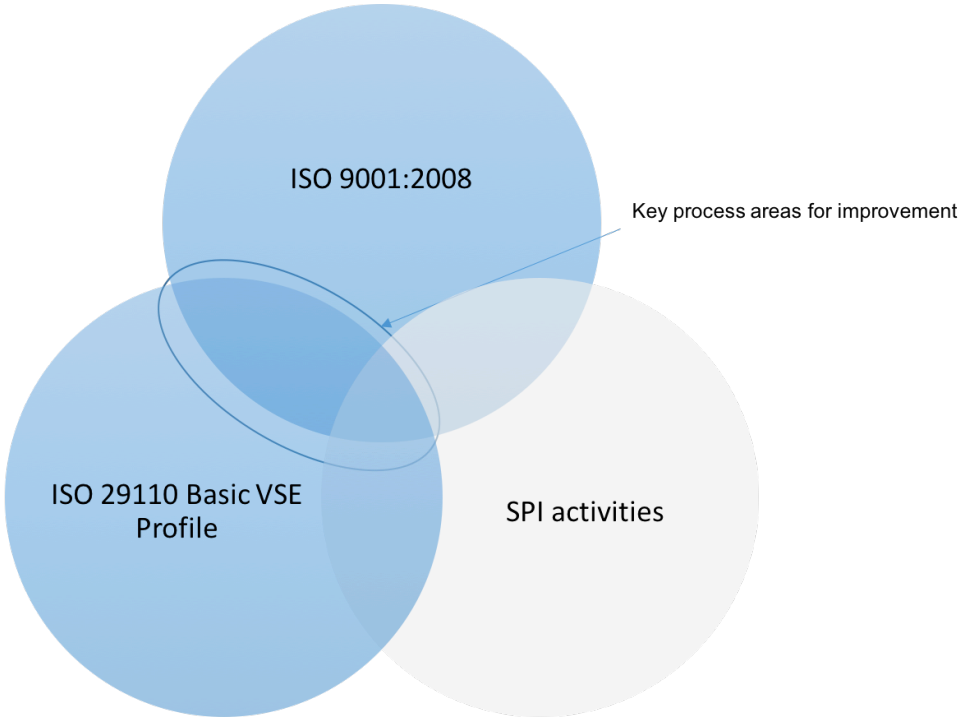

Figure 5. Identifying the key process areas for improvement in a VSE.

The model presented in Figure 5 above is almost identical with the one presented in Figure 1. The focus of this study. However, the model represented here puts the problem into the context of VSEs in general by substituting the case company with ISO 29110 Basic VSE Profile. Since the main (and only) organizational characteristic considered in this study is the size, this can be considered an advantageous development.

8.3 Performing Self-Assessment

The software process assessment was performed by applying the method discussed in Section 8.1.3. The assessment was carried out according to the guidance given in (Varkoi, 2009). Each process of ISO 29110 Basic VSE Profile is assessed based on the collected data documented in Appendix 1 and assessor's knowledge about the software development processes of the case company.

Each process task receives a score in NPLF scale, as described in (Varkoi, 2009) and Table 9 below.

Table 9. The NPLF score system. Reproduced from (Varkoi, 2009).

Process attribute rating values	**Levels of achievement**	**Corresponding percentage scale**
N Not achieved:	There is little or no evidence of achievement of the defined attribute in the assessed process.	0 to 15% achievement
P Partially achieved:	There is some evidence of an approach to, and some achievement of, the defined attribute in the assessed process. Some aspects of achievement of the attribute may be unpredictable.	>15% to 50% achievement
L Largely achieved:	There is evidence of a systematic approach to, and significant achievement of, the defined attribute in the assessed process. Some weakness related to this attribute may exist in the assessed process.	>50% to 85% achievement
F Fully achieved:	There is evidence of a complete and systematic approach to, and full achievement of, the defined attribute in the assessed process. No significant weaknesses related to this attribute exist in the assessed process.	>85% to 100% achievement

The assessment results are documented in Appendix 1. Table 10 below is a snippet from the self-assessment results, presented here as an example.

Table 10. An example of self-assessment results. Adapted from (Varkoi, 2009).

Task	**NPLF**	**Notes**
PM.1.1	**P**	In some projects a vision document is provided and further discussed together with the stakeholders until a consensus is achieved. Official SOWs are not reviewed. Process/task not defined/documented. Tools: MS Word, email

The results table (Table 10) is slightly modified from its original form as it appears in (Varkoi, 2009). The description of the task, input and output are all removed to save space. The omitted details can be found in the ISO 29110 Basic VSE profile

(International Organization for Standardization, 2011b.) by the task identifier found in the Task-column of the assessment results table.

8.4 Analysing Self-Assessment Results

In this section the results of the software processes self-assessment are analysed. The purpose is to analyse the results so that the research question can be partially answered. This section answers the following sub-questions:

- What is the current state of the software processes in the case company?
- What are the key software processes and activities that the case company's improvement efforts should be primarily directed to?

To gain an overall understanding of the maturity of different process activities the results of the assessed task are aggregated according to the selected self-assessment method (Varkoi, 2009). The average score for process activity is derived by aggregating the scores of assessed activity tasks as shown in the example below:

- P,F,F -> average L
- F,L,F -> average F
- N,P,P,L,L -> average P

The results of the self-assessment found in Appendix 1 are aggregated according to the method discussed above. The results of the self-assessment and the derived averages are summarized in Table 11 below.

Table 11. Self-assessment results summarized by process activity.

Process Activity	**NPLF (average)**	**N**	**P**	**L**	**F**
PM.1 Project Planning	P	0	10	3	2
PM.2 Project Plan Execution	P	1	2	3	0
PM.3 Project Assessment and Control	N	3	0	0	0
PM.4 Project Closure	P	0	1	1	0
PM Total	**P**	**4**	**13**	**7**	**2**
SI.1 Software Implementation Initiation	L	0	0	2	0
SI.2 Software Requirements Analysis	P	2	3	1	1
SI.3 Software Architectural and Detailed Design	P	3	2	3	0
SI.4 Software Construction	L	1	1	5	0
SI.5 Software Integration and Tests	P	6	1	3	1
SI.6 Product Delivery	P	4	0	2	0
SI Total	**P**	**16**	**7**	**16**	**3**
Total	**P**	**20**	**20**	**23**	**5**

A summary of the self-assessment results is presented in above. Each row presents a process activity: the identifier and the name of the activity, the average score received the activity and the number of activity tasks with a given score (NPLF). The bolded rows represent the totals for PM-processes, SI-Processes and all processes respectively.

In Table 11 above the key areas for SPI can be identified. The key SPI areas are the process activities and tasks that are not fully achieved (score F). Not a single fully achieved process activity is found in the results. Hence, all assessed process activities are key areas for SPI.

To reiterate, the purpose of this study is to find the key SPI areas and to provide recommendations for SPI. However, it would be valuable to try and prioritize the process activities to be able to provide further recommendations on where to focus the SPI efforts first. Based on the summary presented in Table 11 above, the process activities can be ordered from least mature to most mature. This ordering would serve directly as the priority for SPI. The process activities are presented in the prioritized order in Table 12 below.

Table 12. Process activities prioritized for SPI by maturity.

SPI Priority	Process Activity	NPLF (average)
1	PM.3 Project Assessment and Control	N
2	PM.1 Project Planning	P
3	PM.2 Project Plan Execution	P
4	PM.4 Project Closure	P
5	SI.2 Software Requirements Analysis	P
6	SI.3 Software Architectural and Detailed Design	P
7	SI.5 Software Integration and Tests	P
8	SI.6 Product Delivery	P
9	SI.1 Software Implementation Initiation	L
10	SI.4 Software Construction	L

The SPI priority of process activities presented in Table 12 is based only on the average NPLF-score received by the process activities. No additional prioritizing between activities having the same score is done. However, it would be tempting to try and further prioritize the activities. Further prioritizing could be attempted based on:

- The presumed impact that the activity or task has on quality.
- The presumed effort required to raise the score of a task with respect to the tasks impact to the process.

However, additional prioritizing is not attempted in this study, the attempt is left for future studies and developments. For the purposes of this study it suffices that the key SPI areas have been identified and prioritized.

Regardless of all the conclusions above, arguably the most important finding is the overall score for the software process. It can be seen in Table 11 that the overall score received by the software process is P. According to a DP by Varkoi (Varkoi, 2009) the overall score of P can be interpreted so that the software process is not understood and it needs to be redesigned with competent resources. Further, it can be argued that since the entire software process needs to be completely redesigned identifying the key SPI areas, prioritizing the process activities and deriving any further SPI recommendations is unnecessary.

8.5 Deriving Recommendations for SPI

As discussed in Section 8.4, given the current state of the case company's software process, further analysis and recommendations for SPI to raise the NPLF-score is unnecessary. However, the recommendations are still provided, the presentation and the approach just need to be different.

The original plan was to provide targeted SPI recommendations to raise the score of individual process tasks. To adapt to this new situation, in lieu of the original plan, the SPI recommendations are now rather presented as overall recommendations and things to take into consideration when the software process is being redesigning from the ground up. The method and model for deriving the recommendations stays the same.

After the self-assessment is performed the process for identifying the gaps and deriving the recommendations for SPI is quite straightforward. In the following sections a suitable model for performing the gap analysis and a method for deriving the overall SPI recommendations are discussed and introduced. The gap analysis is performed and based on the results the overall SPI recommendations are derived and presented.

8.6 Method for Deriving Recommendations

The process objectives of ISO 29110 Basic VSE Profile further reference the respective elements of ISO 12207 (see Section 5.2.4 for introduction to ISO 12207). Hence, providing a mapping from the assessed tasks and process objectives to the standard software lifecycle process of ISO 12207.

Furthermore, as discussed in Section 6.3, ISO 90003 provides additional guidance on applying ISO 9001 in the context of software. Like the process objectives of ISO 29110 Basic VSE Profile, it also references the standard lifecycle profiles of ISO 12207 (International Organization for Standardization, 2014). Hence, by cross-referencing ISO 90003 and ISO 29110 Basic VSE Profile, additional guidance to achieving ISO 9001:2008 conformance can be gained.

The mapping from self-assessment results to respective elements of ISO 12207 provides the means for identifying the gaps/shortcomings of the current software process when compared to the ISO 12207 process objectives. Furthermore, the result of the mapping serve as a firm foundation for deriving the concrete SPI recommendations and considerations targeted specifically for the case company.

The recommendations are based on the following elements: Analysts understanding of software processes, self-assessment results and the data it is based on, and corresponding ISO 12207 process objectives. The introduced method has the following characteristics:

- It is based on the results of the self-assessment discussed in Section 8.3 of this study. Hence, it takes into account the currents state of the software processes of the case company.
- It provides a mapping from the assessed process activities to the standardised software lifecycle processes and process objectives of ISO 12207. Hence, the gaps and shortcomings of the current processes are identified and a suitable foundation for deriving the recommendations and/or considerations is formed.
- It relies largely on the analyst's understanding of the problem and solution which could be considered a negative aspect. However, in the context of this study this is acceptable.

It can be argued that the introduced method is slightly vague, since it relies largely on the analyst knowledge. However, objectivity and repeatability are not key factors of this study. A more structured method is not introduced nor applied. Thus, the method discussed here is suitable for the needs of this study and it is selected as the method for identifying the gaps and deriving the recommendations.

8.7 Developing Model for Identifying Gaps

To be able to provide the overall recommendations for the software redesigned process the gaps need to be identified first. The model for identifying the gaps of the case company's software processes with respect to ISO 9001 requirements is presented in Figure 6 below.

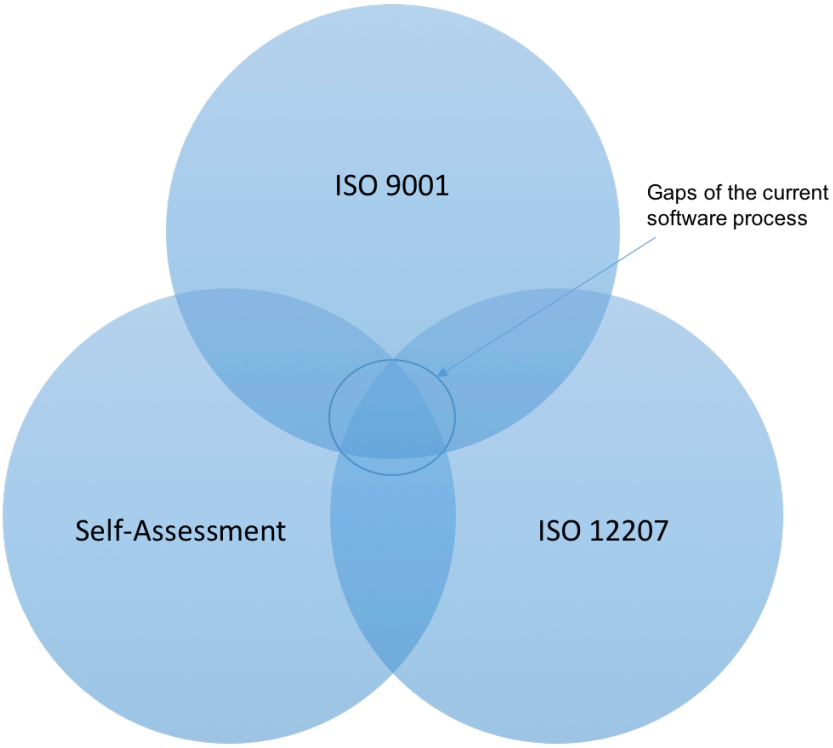

Figure 6. Gaps of the current software processes with respect to ISO 9001.

The model presented in Figure 6 elicits how ISO 9001 requirements, the current state of the case company's software process and the software lifecycle processes of ISO 12207 are all interrelated. By applying the presented model, the gaps of the current software process of the case company with respect to ISO 9001 requirements can be identified. The model presented in Figure 6 is almost identical with the one presented in Figure 5. Identifying the key process areas for improvement in a VSE. However, the model represented here grounds the problem firmly back into the context of the case company.

8.8 Performing Gap Analysis

The gap analysis is performed according to the method and model discussed and introduced earlier in this section. The results are documented in a table in Appendix 2. A snippet from the results is presented in Table 13 below, as an example.

Table 13. Example of gap analysis results.

ISO 12207	Current State	Shortcomings
6.3.1 Project Planning Process • The scope of the work for the project is defined • The tasks and resources necessary to complete the work are sized and estimated • Plans for the execution of the project are developed • Plans for the execution of the project are activated.	PM.1 The scope of some of the projects is currently estimated, but in ad-hoc basis. The situation is similar with the project plans. The resource needs are never estimated.	• The process for estimating the scope is not defined • Recourses are not considered • Process for executing the projects is not defined

An example of gap analysis results is presented in Table 13 above. The left most column contains the elements of ISO 12207 software lifecycle profiles referenced by ISO 29110 Basic VSE Profile. Centre column shows the respective process activities of the profile with a summary of the assessment results. The right most column contains the identified gaps.

The gap analysis is performed according to the model introduced in the previous section. First process objectives of each process activity are mapped to be able to identify the elements of ISO 12207 associated with each process activity. Second the ISO 12207 elements are compared to the assessment data and assessment results of each corresponding process activity, and finally, the shortcomings are identified.

8.9 Analysing Results of Gap Analysis

In this section the results of the gap analysis are analysed and overall recommendations are provided to be considered when the case company's software processes are redesigned. The provided recommendations are largely based on common knowledge

and the analysts understanding about software processes in general. Hence, for the most part citations are not provided.

When analysing the results, it is quickly recognized that the main problem is that none of the processes are defined. Some of the process tasks are actually performed at some level, some even fully achieved, but there are no established strategies nor defined processes.

By adopting a suitable SDLC model and by integrating supporting processes many advantages could be gained. In Section 1 it was discussed how agility is one of the main competitive advantages of the case company. Hence, an agile SDLC model should be considered. Many agile models, for example Scrum, are inadequate since they only partly cover the entire SDLC. They ignore other important phases like visioning or requirements elicitation and delivery.

The adoption of DAD (*Disciplined Agile Delivery*) framework should be considered. It is a framework that extends the existing agile SDLC models and covers the full delivery lifecycle, from visioning to delivery (Disciplined Agile Consortium, no date). Figure 7 below illustrates the high level DAD lifecycle.

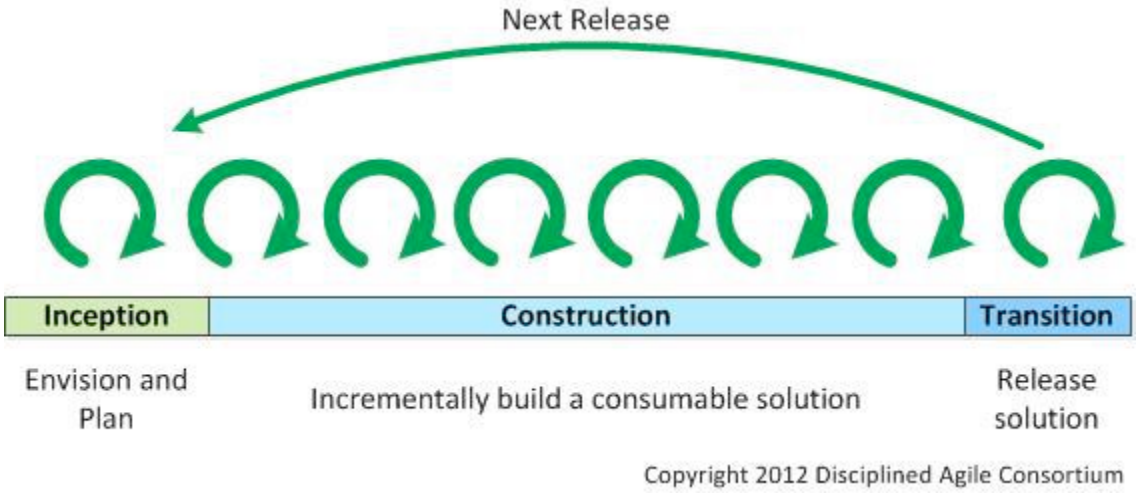

Figure 7. High level DAD lifecycle. Reprinted from (Disciplined Agile Consortium, no date).

Figure 7 above illustrates the high level DAD lifecycle. DAD lifecycle covers the full lifecycle from visioning to deployment. From the high level presentation in Figure 7 it might be difficult see how the adoption of such model will address the shortcomings identified in gap analysis. Hence, in the following sub-sections an example process

based on the DAD framework is discussed in more detail. The example will also serve as the overall SPI recommendation that should be considered when the software process is redesigned. It should be noted that provided example is not a complete process by any means.

8.9.1 Inception

At the inception phase the product/project is envisioned and the scope of the project is defined. Initial requirements are created, project feasibility is assessed and risks are identified. Furthermore, required resources are assessed and an initial schedule is created. Traceability record is created and traceability from vision to requirements is recorded and basic configuration management practises are established and initialized.

8.9.2 Construction

At the construction phase the project is constructed in an agile fashion. The requirements are being further specified, designed and risks assessed in a JIT manner before each iteration in a planning event. The testability of requirements is ensured, acceptance criteria is defined and test are created. The requirements are split into user stories and tasks which are estimated and assigned to team members. Traceability from requirements to stories and tasks and further to tests is recorded. Spot tests by QA (*Quality Assurance*) and unit test by developers are performed continuously, and user documentation is updated while the requirements are being implemented. The software configuration is continuously updated as new items are being built. Daily work is continuously being tracked and reported by burn down charts and in daily meetings. The work is being constantly reviewed by the stakeholders in demo events at the end of each iteration. Each iteration is concluded with a meeting where the iteration is assessed and the process is continuously improved.

8.9.3 Transition

At the transition phase the acceptance tests are executed, defects are fixed and the version is deployed to the production. The software configuration is updated and baselined, the configuration includes everything from vision and traceability record to user documentation. QA team reports defects which are investigated, classified and fixed

accordingly. The QA team performs regression testing, acceptance testing and additional exploratory tests. User documentation is also verified. Once the QA team accepts the version, it is delivered to the customer for acceptance. An additional pilot phase could also be included. Once the version is accepted by all stakeholders it is deployed to the production.

8.10 Summary and Conclusions

In this section a method for performing the self-assessment on the case company's software processes was selected and a model for identifying the key SPI areas was developed. The self-assessment was performed and the results were analysed. The result of the analysis is crude, it was identified quite early on that the current state of the software processes is unsatisfactory and any further analysis are unnecessary. The software process needs to be completely redesigned from the ground up.

Regardless of the conclusion reached, the recommendations for SPI were derived. However, the approach needed to be different from what was originally planned. A model for performing the gap analysis was introduced and applied. Based on the identified gaps an overall SPI recommendation was provided as an example process. The provided example process should be considered when the case company redesigns its software processes.

9 Discussion and Final Conclusions

The purpose of this study was to identify the key SPI areas of the case company based on the ISO 9001:2008 requirements and to provide SPI recommendations to address the identified shortcomings or gaps. It was clear from the beginning that to achieve the purpose the following steps would be necessary:

1. Collect data on the case company's current software process
2. Assess the software process based on the collected data and ISO 9001:2008 requirements
3. Based on the results of the assessment, identify the gaps/shortcomings of the current software process in contrast to ISO 9001:2008 requirements

4. Provide recommendations for SPI to address the identified shortcomings

The data collection phase was performed according to the plan, by observing and participating in the software development and maintenance activities in the case company. No interviews or other additional data collection methods were applied and the data was mainly collected already in 2014. Thus, it could be argued that the results of this study are not reliable nor valid. However, given the small size of the company and the fact that the author has engaged fulltime to its software activities for a number of years and in multiple roles, the data can be considered valid and reliable.

The first sections (Sections 3 - 5) of this study discussed the necessary background knowledge required to understand the problem. Quality was discussed on a general level and in the contexts of SMEs and software. SPI was briefly discussed and the most important SPI frameworks were introduced and compared. An adequate understanding of quality, for the purposes of this study, was gained. However, it would have been interesting and beneficial to study SPI methods and SPI frameworks more deeply.

A sound understanding on the foundations that the quality frameworks, including ISO 9001:2008, are built on was gained in the beginning and in Section 6 ISO 9001:2008 was introduced. The evolution of ISO 9001 was discussed from the software perspective, the requirements were discussed, and finally, its application in the context of software was discussed.

Adequate understanding about SMEs, software quality and ISO 9001:2008 and its application to software was gained. The latter sections (Sections 7 - 8) built upon this knowledge and the methods and models for performing the assessment and deriving the recommendations for SPI were introduces and applied. The results were analysed and discussed, and finally, the SPI recommendations were presented.

The results of the assessment were quite crude and surprising. It was originally planned that based on the assessment results targeted SPI recommendations for each process area could have been provided. Instead, the assessment results indicated that the entire software process needs to be redesigned from the ground up. The selected self-assessment method was suitable for the purposes of this study. However, it would

have been interesting to see the results if a suitable method for performing a formal and repeatable capability assessment would have been found and applied.

If the results of the self-assessment had been different, it would have been interesting to try and further prioritize the identified key SPI areas. For example, the knowledge gained on quality could have been applied to prioritize the assessed process tasks based on their estimated impact on quality. Furthermore, the assessed task could have been scored by the estimated work effort required to address the shortcomings with respect to the tasks impact on the overall software process. Many other ways to further prioritize the key SPI areas could probably have been developed and studied.

As discussed earlier the original plan was to provide SPI recommendations targeted to address the shortcomings of a given process area. Since the assessment results indicated that the entire software process needs to be redesigned, the plans had to be adapted in the very final stage of the study. As a result, an overall example software process was presented that the case company should consider when the software process is redesigned. The presented example process addressed many, if not all, of the shortcomings of the assessed process. However, it would have been interesting to study the overall process in more detail. Furthermore, it would have been beneficial to provide a comparison between the assessed and the recommended example process to see how the shortcomings are addressed.

After 2014 and during this study a lot has happened in the case company and many SPI actions have already been taken. However, many lessons were learned in this process and the knowledge gained and the methods and models that were developed can be used for SPI purposes of the case company in the future. After all, quality is about continuous improvement.

To reiterate, the purpose of this study was to identify the key SPI areas of the case company based on ISO 9001:2008 requirements and to provide SPI recommendations to address the identified shortcomings or gaps. The research question this study tried to answer was: “Based on ISO 9001:2008 and the current state of the case company, what are the key software process areas for improvement and how to improve?” and the sub-questions:

- What is the current state of the software processes in the case company?

- What are the key software processes and activities that the case company's improvement efforts should be primarily directed to?
- For the key improvement areas, what are some of the possible SPI actions to take?

The study achieved its purpose and answered the research questions. However, the results were not quite what was originally expected and some adjustments to the plans had to be made at the very final stages of the study. The final conclusion of this study is that the software process of the case company needs to redesigned completely and that the provided example process should be considered when doing so.

References

Bath, S., 2010. *Total Quality Management.* Mumbai: Himalaya Publishing House.

Chemuturi, M., 2014. *Mastering Software Quality Assurance.* Ft. Lauderdale: Ross Publishing Inc..

Dale, B., van der Wiele, T. & van Iwaardeen, J., 2013. *Managing Quality.* 5th Edition ed. Somerset: Wiley-Blackwell.

Disciplined Agile Consortium, n.d. *Disciplined Agile 2.0.* [Online]
Available at: http://www.disciplinedagiledelivery.com
[Accessed 5 2016].

Dybå, T., Dingsoyr, T. & Moe, N. B., 2004. *PROCESS IMPROVEMENT IN PRACTICE.* Hingham: Kluwer Academic Publishers.

European Union, 2015. *User guide to the SME Defnition,* Luxembourg: Publications Office of the European Union.

Foegen, M. & Richter, J., 2003. *The University of Kansas.* [Online]
Available at:
http://people.eecs.ku.edu/~saiedian/Teaching/Fa15/810/Readings/cmmi.pdf
[Accessed 05 2016].

Horch, J., 2003. *Practical Guide to Software Quality Management.* Norwood: Artech House Books.

International Organization for Standardization, 2008. *ISO/IEC 12207:2008(en) Systems and software engineering — Software life cycle processes.* [Online]
Available at: https://www.iso.org/obp/ui/#iso:std:iso-iec:12207:ed-2:v1:en
[Accessed 5 2016].

International Organization for Standardization, 2008. *Quality management systems - Requirements.* ISO 9001:2008 ed. Geneva: ISO/IEC.

International Organization for Standardization, 2011. *Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 1: Overview,* Geneva: ISO/IEC.

International Organization for Standardization, 2011. *Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile,* Geneva: ISO/IEC.

International Organization for Standardization, 2014. *ISO/IEC 90003:2014(en) Software engineering — Guidelines for the application of ISO 9001:2008 to computer software.* [Online]
Available at: https://www.iso.org/obp/ui/#iso:std:iso-iec:90003:ed-2:v1:en
[Accessed 5 2016].

Juran, J. M., 1988. *Quality Control Handbook.* New York: McGraw-Hill.

Kenett, R. & Baker, E., 1999. *Software Process Quality.* Baton Rouge: CRC Press.

Letze, S. R., Zairi, M. & Whymark, J., 1997. *TQM – Fad or Tool for Sustainable Competitive Advantage? An Empirical Study of the Impact of TQM on Bottom Line Business Results,* Bradford: s.n.

Mutafelija, B. & Stromberg, H., 2003. *Systematic Process Improvement using ISO 9001.* Norwood: Artech House Books.

Oskarsson, Ö., 1999. *An ISO 9000 Approach to Building Quality Software.* [Online]
Available at: http://www.oskarsson.se
[Accessed 04 2016].

Oxford University Press, n.d. [Online]
Available at: http://www.oxfordjournals.org/our_journals/tropej/online/ce_ch14.pdf
[Accessed 5 2016].

Robitaille, D., 2010. *ISO 9001 : 2008 for Small and Medium-Sized Businesses.* 2nd Edition ed. Milwaukee: ASQ Quality Press.

Software Engineering Institute, Carnegie Mellon University, 1993. *Capability Maturity Model for Software, Version 1.1.* CMU/SEI-93-TR-024 ed. Pittsburgh: CMU.

Software Engineering Institute, Carnegie Mellon, 2010. *CMMI® for Development, Version 1.3.* CMU/SEI-2010-TR-033 ed. Pittsburgh: CMU.

Suryn, V., Hailey, V. & Coster, A., 2004. Huge potential user base for ISO/IEC 90003. *ISO Management Systems,* 4(4), pp. 34-39.

The British Standards Institution, n.d. *bsigroup.* [Online]
Available at: http://www.bsigroup.com/LocalFiles/zh-cn/pdf/Whitepaper-the-history-and-future-of-ISO-9001.pdf
[Accessed 5 2016].

Völcker, C. et al., 2002. *SPiCE in Action Experiences in Tailoring and Extension.* Bonn, GI, Gesellschaft für Informatik.

Varkoi, T., 2009. *Deployment Packages for the Generic Profile Group for VSEs Developing Systems and/or Software.* [Online]
Available at: http://profs.etsmtl.ca/claporte/english/VSE/Deploy-Pack/DP-Self-Assessment-V1.2.doc
[Accessed 5 2016].

Varkoi, T. & Mäkinen, T., 2010. *A Process Assessment Model for Very Small Software Entities,* Pori: Tampere University of Technology, CoSE.

Whittington, L., n.d. *The 9000 Store.* [Online]
Available at: http://the9000store.com/downloads/standards/ISO-9001-Requirements-Explained.pdf
[Accessed 05 2016].

Yan, B. & Zhang, L., 2011. *An approach of quality management in the small business environment of South Africa.* Singapore, IEEE, pp. 684-688.

Appendix 1: Self-Assessment Results and Data

Task	NPLF	Notes
PM.1.1	**P**	In some projects a vision document is provided and further discussed together with the stakeholders until a consensus is achieved. Official SOWs are not reviewed. Process/task not defined/documented. Tools: MS Word, email
PM.1.2	**P**	In most of the projects the delivery methods are agreed together with the stakeholders. Process/task not defined. Tools: MS Word, email
PM.1.3	**L**	All projects are always split into tasks. Task are recorded in the issue tracker. Tasks are sometimes reviewed with stakeholders. Tools: JIRA
PM.1.4	**F**	Task are always estimated in detail. Tools: JIRA
PM.1.5	**P**	Only a single software team exists in the company, which is typically 100% committed one project. The materials, tools, training etc. are always defined in JIT basis. Process/task not defined/documented.
PM.1.6	**L**	The entire team works on a single project and the roles of the team are static. Process/task not defined/documented.
PM.1.7	**L**	The tasks are worked on prioritized order and sequences/dependencies are resolved in JIT basis. No dates, other then the release date for the entire project, are assigned. Tasks are always assigned to team members. Process/task not defined/documented. Tools: JIRA
PM.1.8	**P**	With customer projects the costs are always calculated and documented. With internal projects not. Process/task not defined/documented. Tools: JIRA, MS Word
PM.1.9	**P**	In some projects risks are identified and documented. Process/task not defined/documented. Tools: MS Word
PM.1.10	**P**	Version control strategy is the same with all projects. Not documented in the project plan.
PM.1.11	**P**	In some projects an official project plan is created. Process/task not defined/documented. Tools: MS Word
PM.1.12	**P**	In some projects these are considered, but not included in the project plan.

		Process/task not defined/documented. Tools: MS Word
PM.1.13	**P**	If a project plan is created it is balloted until a consensus is achieved between the stakeholders. Official verification is not conducted. Process/task not defined/documented. Tools: email
PM.1.14	**P**	If a project plan is created it is balloted until a consensus is achieved between the stakeholders. Official verification is not conducted. Process/task not defined/documented. Tools: email
PM.1.15	**F**	All projects items are under version control. Process/task not defined/documented. Tools: SVN, Artifactory
PM.2.1	**L**	The project tasks are recorded in issue tracker and their statuses are updated daily. At anytime progress of the project can monitored with the issue tracker. Process/task not defined/documented. Tools: JIRA
PM.2.2	**P**	In some projects change request are analysed and evaluated and negotiated with stakeholders. Change requests are not tracked, but tasks and requirements are directly updated. All this is done in ad-hoc basis and requires more control. Process/task not defined/documented. Tools: JIRA, MS Word
PM.2.3	**N**	Such meeting are not conducted, since the entire team is co-located and communication is frequent. Process/task not defined/documented.
PM.2.4	**P**	Such official meetings are not conducted, but change request are received and handled by other methods as in PM.2.2. Process/task not defined/documented.
PM.2.5	**L**	Backups are in place for the entire development environment. However, these need to be documented. Process/task not defined/documented. Tools: OS backup tools, cloud services, RAID
PM.2.6	**L**	Recovery has never been tried. This should be exercised. Process/task not defined/documented.
PM.3.1	**N**	Project progress is typically evaluated only when projects is (or is in a risk) overdue. Process/task not defined/documented. Tools: JIRA, MS Word
PM.3.2	**N**	Actions for correcting the deviations are established, but in ad-hoc basis. Correction register does not exist, all changes are done directly into the tasks and project plan.

		Process/task not defined/documented.
PM.3.3	**N**	Change request are not tracked. All changes are done directly to tasks and project plan. Process/task not defined/documented.
PM.4.1	**P**	Projects are not officially accepted/signed. Some projects have an UAT or pilot phase which can be considered as acceptance. Process/task not defined/documented.
PM.4.2	**L**	Project repository is always up to date, even though not officially accepted. Tools: SVN, Artifactory
SI.1.1	**L**	Project plan is reviewed with the team. Process/task not defined/documented. Tools: email, MS Word
SI.1.2	**L**	The development environment is always updated in JIT basis. Process/task not defined/documented.
SI.2.1	**F**	Task are always assigned to team members. Tools: JIRA
SI.2.2	**L**	In some project a separate requirements specification document is created while others just have task in the issue tracker. Technical feasibility and scope analysed in JIT basis, and required changes are made directly into the requirements. Process/task not defined/documented. Tools: JIRA, MS Word
SI.2.3	**P**	Requirements are balloted until a consensus is achieved amongst the stakeholders, but are not officially approved. The findings are not documented into a verification result, but changes are made directly into the requirements. Process/task not defined/documented. Tools: MS Word, JIRA
SI.2.4	**P**	Requirements are balloted until a consensus is achieved amongst the stakeholders, but are not officially approved. The findings are not documented into a validation result, but changes are made directly into the requirements. Process/task not defined/documented. Tools: Ms Word, JIRA
SI.2.5	**N**	The user manuals are produced in ad-hoc basis. More control is required. Process/task not defined/documented. Tools: MS Word
SI.2.6	**N**	The user manuals are produced in ad-hoc basis. More control is required. Process/task not defined/documented. Tools: MS Word
SI.2.7	**P**	If a separate requirements specification is produced it is not included in the configuration. Neither is the user documentation.

		However, requirements in issue tracker are. Process/task not defined/documented. Tools: JIRA, SVN, Artifactory
SI.3.1	**L**	Separate design tasks are not created. If a requirement is recorded in issue tracker it is assumed to cover all tasks. At different stages the task is assigned to required team member. Process/task not defined/documented. Tools: JIRA
SI.3.2	**L**	Requirements are discussed in daily basis and are fully understood by all team members. Process/task not defined/documented.
SI.3.3	**P**	Software design is sometimes documented with mock-ups and interfaces. Rarely are any architectural design documents created. A traceability record is not created/updated. Process/task not defined/documented. Tools: Wiremock, MS Word
SI.3.4	**P**	The design is balloted until a consensus is achieved amongst the stakeholders. The verification of the design happens on ad-hoc basis and is never officially approved. Verification and/or traceability records are not created/updated. Process/task not defined/documented.
SI.3.5	**L**	Unit and system test are created and executed. Process/task not defined/documented. Process/task not defined/documented. Tools: Robot framework, Junit
SI.3.6	**N**	Tests are not approved by anyone.
SI.3.7	**N**	Traceability record is not created/updated. Tests are
SI.3.8	**N**	Software design, traceability record, test cases and procedures are not currently part of the software configuration/baseline.
SI.4.1	**L**	Separate construction tasks are not created. If a requirement is recorded in issue tracker it is assumed to cover all tasks. At different stages the task is assigned to required team member. Process/task not defined/documented. Tools: JIRA
SI.4.2	**L**	Design is discussed in daily basis and is fully understood by all team members. Process/task not defined/documented.
SI.4.3	**L**	Task are linked in the issue tracker directly into the respective components/modules. Process/task not defined/documented.
SI.4.4	**L**	Unit tests are created with adequate coverage. Process/task not defined/documented.
SI.4.5	**L**	All defects found in the construction phase are normally corrected immediately. Process/task not defined/documented.

SI.4.6	**N**	Traceability record is not created/updated. Components are not incorporated.
SI.4.7	**P**	Software components are part of the configuration/baseline. Traceability record does not exist. Process/task not defined/documented.
SI.5.1	**L**	Separate integration and testing tasks are not created. If a requirement is recorded in issue tracker it is assumed to cover all tasks. At different stages the task is assigned to required team member. Process/task not defined/documented. Tools: JIRA
SI.5.2	**L**	Test cases and procedures are fully understood by the tester. She also ensures that the test environment is always up to date. Process/task not defined/documented.
SI.5.3	**P**	Software is integrated and test cases and procedures are updated as required. However, more control is required. Process/task not defined/documented. Tools: Robot Framework
SI.5.4	**F**	System tests are performed for integrated software and results are reported. Process/task not defined/documented. Tools: Robot Framework
SI.5.5	**L**	Defects are normally corrected immediately before release. After corrections software is tested again. However, regression testing should be defined more clearly. Process/task not defined/documented. Tools: JIRA, Robot Framework
SI.5.6	**N**	Traceability record is not created/updated.
SI.5.7	**N**	Operation guide and other user documentation is updated only on ad-hoc basis. Process/task not defined/documented.
SI.5.8	**N**	Operation guide and other user documentation is updated only on ad-hoc basis. Hence, they are not verified nor approved. Process/task not defined/documented.
SI.5.9	**N**	User documentation is updated only on ad-hoc basis. Process/task not defined/documented.
SI.5.10	**N**	User documentation is updated only on ad-hoc basis. Hence, it is never verified/approved. Process/task not defined/documented.
SI.5.11	**N**	Test cases and procedures, traceability record, test report, operation and user guides are only partly included in some configuration. Mostly not. Process/task not defined/documented.
SI.6.1	**N**	Delivery tasks are not considered. Process/task not defined/documented.

SI.6.2	**L**	Software configuration is understood by the team members.
SI.6.3	**N**	Maintenance documentation is not created/updated.
SI.6.4	**N**	Maintenance documentation is not created/updated.
SI.6.5	**N**	Maintenance documentation is not created/updated.
SI.6.6	**L**	Delivery is performed according to the delivery instructions, if such exists.

Appendix 2: Gap Analysis Results

ISO 12207	Current State	Shortcomings
6.3.1 Project Planning Process • the scope of the work for the project is defined • the tasks and resources necessary to complete the work are sized and estimated • plans for the execution of the project are developed; and • plans for the execution of the project are activated.	PM.1 The scope of some of the projects is currently estimated, but in ad-hoc basis. The situation is similar with the project plans. The resource needs are never estimated.	• The process for estimating the scope is not defined • Recourses are not considered • Process for executing the projects is not defined
6.3.7 Measurement Process • the information needs of technical and management processes are identified.	PM.1 The needs are identified in ad-hoc / JIT basis.	• The processes are not defined
6.3.2 Project Assessment and Control Process • progress of the project is monitored and reported • actions to correct deviations from the plan and to prevent recurrence of problems identified in the project, are tak-	PM.2, PM.3, PM.4 Sometimes the progress is monitored and reported, but this typically happens only when the project is in risk to become overdue.	• The process for executing the projects is not defined • The process of monitoring the progress of projects is not defined • The project objectives are not recorded

en when project targets are not achieved; and • project objectives are achieved and recorded.		
6.3.7 Measurement Process • the required data are collected, stored, analyzed, and the results interpreted; and • information products are used to support decisions and provide an objective basis for communication.	PM.2, PM.3, PM.4 Very limited metrics are collected. Decisions or communications cannot be based on these.	• Metrics are not established • Limited issue and task types to allow meaningful conclusions
6.4.8 Software Acceptance Support Process • the product is completed and delivered to the acquirer	PM.2, PM.3, PM.4 The products are tested and delivered, but on ad-hoc basis.	• The delivery process is not defined
7.2.8 Software Problem Resolution Process • problems are recorded, identified and classified; and • problems are tracked to closure.	PM.2, PM.3, PM.4 Problems are recorded and tracked to closure, but on ad-hoc basis. No identification or classification.	• Incident management process is not defined
7.1.2 Software Requirements Analysis Process • changes to the software requirements are evaluat-	PM.2 Changes are evaluated, but not tracked, but on ad-hoc basis. All changes are done directly to tasks and	• Change management process is not defined

ed for cost, schedule and technical impact.	other items.	
7.2.6 Software Review Process • management and technical reviews are held based on the needs of the project • review results are made known to all affected parties • action items resulting from reviews are tracked to closure.	PM.2 Official review meetings are not held.	• Official reviews are not organized
6.3.4 Risk Management Process • risks are identified as they develop and during the conduct of the project	PM.1, PM.2 In some projects risks are documented, but in ad-hoc basis.	• The risk management process is not defined
7.2.6 Software Review Process • risks and problems are identified and recorded.	PM.1, PM.2 In some projects risks are documented, but in ad-hoc basis.	• The risk management process is not defined
7.2.2 Software Configuration Management Process • a software configuration management strategy is developed • items generated by the process or project are identified,	PM.1 Configuration management is well performed, but not documented.	• The configuration management strategy should be documented.

defined and baselined • modifications and releases of the items are controlled • modifications and releases are made available to affected parties • the storage, handling and delivery of the items are controlled.		
7.2.3 Software Quality Assurance Process • a strategy for conducting quality assurance is developed • evidence of Software quality assurance is produced and maintained • problems and/or non-conformance with requirements are identified and recorded; and • adherence of products, processes and activities to the applicable standards, procedures and requirements are verified.	PM.1, PM.2 Quality assurance is performed at some level. Products are always tested and sometimes piloted or accepted by the customer. However, there is no documented process and this takes place in ad-hoc basis.	• Software development and maintenance processes are not defined • QA processes are not defined and integrated
6.4.1 Stakeholder Requirements Definition Pro-	SI.1, SI.2 Requirements are defined,	• Requirements elicitation process is not defined

cess • the required characteristics and context of use of services are specified.	but on ad-hoc basis.	
7.1.2 Software Requirements Analysis Process • the requirements allocated to the software elements of the system and their interfaces are defined • software requirements are analyzed for correctness and testability • the software requirements are approved and updated as needed; and • the software requirements are baselined and communicated to all affected parties.	SI.1, SI.2 The requirements are balloted until a consensus is achieved. Requirements are not officially approved or baselined.	• Requirements analysis process is not defined • Requirements baselines are not established
7.1.3 Software Architectural Design Process • a software architectural design is developed and baselined that describes the software items that will implement the software requirements • internal and external interfaces of	SI.3 Software architecture is not typically designed. No traceability record is maintained.	• Software development process is not defined • Software design process is not defined and integrated • Traceability strategy is not established

each software item are defined • and consistency and traceability are established between software requirements and software design.		
7.1.4 Software Detailed Design Process • a detailed design of each software component, describing the software units to be built, is developed • external interfaces of each software unit are defined; and • consistency and traceability are established between the detailed design and the requirements and architectural design.	SI.3 Some design is done e.g. with mock ups. This is done in ad-hoc basis. No traceability record is maintained.	• Software development process is not defined • Software design process is not defined and integrated • Traceability strategy is not established
7.1.5 Software Construction Process • verification criteria are defined for all software units against their requirements • software units defined by the design are produced • consistency and	SI.4 Unit tests are developed and applied. Traceability records are not maintained.	• Software development and maintenance processes are not defined • Unit testing strategy is not established • Traceability strategy is not established

traceability are established between software units and requirements and design; and • verification of the software units against the requirements and the design is accomplished.		
7.1.6 Software Integration Process • software items are verified using the defined criteria • software items defined by the integration strategy are produced • results of integration testing are recorded • consistency and traceability are established between software design and software items;	SI.5 System tests are always performed and results are reported. However, process is not well defined or documented. No traceability record exists.	• The strategy for establishing acceptance criteria is not defined • Traceability strategy is not established •
7.1.7 Software Qualification Testing Process • criteria for the integrated software is developed that demonstrates compliance with the software requirements • integrated software is verified using the	SI.5 System tests are always performed and results are reported.	• The strategy for establishing acceptance criteria is not defined • The process for software acceptance testing is not defined •

defined criteria; and • test results are recorded.		
6.1.2 Supply Process • a product and/or service that meets the agreed requirements are developed by the supplier • the product and/or service is delivered to the acquirer in accordance with the agreed requirements; and • the product is installed in accordance with the agreed requirements.	SI.2, SI.3, SI.4, SI.5, SI.6 The products are developed and delivered according to requirements. However, this is done in ad-hoc basis.	• The software design process is not defined • The software development and maintenance processes are not defined • The delivery process is not defined •
7.2.1 Software Documentation Management Process • a strategy identifying the documentation to be produced during the life cycle of the software product or service is developed • documentation to be produced by the process or project is identified; and • documentation is developed and made available in accordance with	SI.2, SI.3, SI.4, SI.5, SI.6 All documentation is done in ad-hoc basis.	• The software development and maintenance processes are not defined • The documentation strategy is not defined and integrated

identified standards.		
7.2.4 Software Verification Process • a verification strategy is developed and implemented • criteria for verification of all required software work products is identified • required verification activities are performed • defects are identified and recorded; and • results of the verification activities are made available to the customer and other involved parties.	SI.3, SI.4, SI.5, SI.6 Defects are recorded and fixed during the construction. This is done in ad-hoc basis no documented strategy exists.	• The software development and maintenance processes are not defined • The verification strategy is not established and integrated
7.2.5 Software Validation Process • a validation strategy is developed and implemented • criteria for validation of all required work products are identified • required validation activities are performed • problems are identi-	SI.3, SI.4, SI.5, SI.6 No process for validation exists. Requirements are discussed continuously and adjustments are made, but this is done in ad-hoc basis.	• The software development and maintenance processes are not defined • The validation strategy is not established and integrated

fied and recorded • results of the validation activities are made available to the customer and other involved parties.		