

Timo Mulju

# Improvement of a Technical Solution Process in a Software Company

---

Helsinki Metropolia University of Applied Sciences

Master's Degree

Industrial Management

Master's Thesis

19 April 2013

Author Title	Timo Mulju Improvement of a Technical Solution Process in a Software Company
Number of Pages Date	65 pages + 2 appendices 19 April 2013
Degree	Master's degree
Degree Programme	Industrial Management
Instructor	Thomas Rohweder, DSc(Tech), Principal Lecturer
<p>This Thesis presents an improved process for Technology Solutions in the case company, which is a mid-sized software company located in Finland. Technology Solutions is a process area, which concentrates on technology selections and software design. There are many existing frameworks for different processes that this Thesis utilizes in the improvement of the model. The current situation is that the case company does not have a defined process, just different procedures and document templates.</p> <p>This study develops the process through action research. The Thesis presents the first version of the model, which is then iteratively improved in validating projects. The data used is collected by conducting a companywide inquiry, several interviews and a validating session. The interviewees were selected among actual people using the process; Lead Developers and Technical Architects of the case company. By combining findings from the literature and collected data, the first version of the model was created, which was then piloted in the validating projects. The outcome of this Thesis is this new, more defined Technology Solutions process.</p> <p>This study has value for the case company since it enables a totally new process for software design. The process itself enables a common way to design software, knowledge sharing and a more agile way to operate.</p>	
Keywords	Technology Solutions, process improvement, CMMI

## Contents

Preface

Abstract

Table of Contents

List of figures

List of tables

Acronyms

1	Introduction	1
1.1	Case Company	4
1.2	Business Problem and Objective	7
2	Research Approach	9
2.1	Research Methods and Process	9
2.2	Data Collection and Analysis Methods	12
2.3	Reliability and Validity	14
3	Best Practice for Process Management	16
3.1	CMMI for Development	17
3.2	TSP and PSP	21
3.3	Lean Methods	23
3.4	Agile Methods and Architecture Driven Design	25
3.5	Six Sigma	28
3.6	Objective of Knowledge Sharing	29
3.7	Conceptual Framework of Process Design for Technical Solutions	32
4	Analysis of the Data	36
4.1	Approach	36
4.2	The Perspective of All the Staff: Inquiry	36
4.3	Lead Developer Perspective: Interviews	39
4.4	Summary	41
5	Building the Technical Solutions Process	42
5.1	Analysis of the Stakeholder Perspectives	42
5.1.1	Management Perspective	44
5.1.2	Project Manager Perspective	45



5.1.3	Lead Developer	45
5.1.4	Developer or Expert Perspective	46
5.1.5	Customer Perspective	46
5.2	Proposal for a New Process	46
5.2.1	Preliminary Design	48
5.2.2	Architecture Design	51
5.2.3	Detailed Design	53
5.2.4	Development	56
5.2.5	Knowledge Sharing and Enabling the Learning Organization	56
6	Implementation of the new process	60
7	Discussion and Conclusions	61
7.1	Summary	61
7.2	Practical Implications	62
7.3	Evaluation	62
7.3.1	Outcome Compared to Objective	62
7.3.2	Reliability and Validity in this Study	63
	References	65
	Appendices	
	Appendix 1. Questions in the inquiry	
	Appendix 2. Questions in interviews	



## LIST OF FIGURES

Figure 1. Holistic view of the role of process.

Figure 2. Lewin's Action Research cycle.

Figure 3. Research process designed for this thesis.

Figure 4. CMMI model components.

Figure 5. Structure in PSP and TSP.

Figure 6. Development model in agile methods.

Figure 7. Development model in Architecture centric model.

Figure 8. Architecture of the software engineering experience environment.

Figure 9: Conceptual framework.

Figure 10. Basic Technical Solution process integrated from CMMI Technical Solution process area.

Figure 11. Timeline in simplified, typical project.

Figure 12. Proposed process.

Figure 13. Detailed design advancing.

## LIST OF TABLES

- Table 1. Details of the data collection.
- Table 2. Capability (0-3) and Maturity (4-5) levels in CMMI.
- Table 3. Aspects in Preliminary design phase.
- Table 4. Selection criteria when comparing alternatives.
- Table 5. Aspects in architectural plan.
- Table 6. Aspects in the detailed planning.
- Table 7. Suggested Wiki structure.
- Table 8. Challenges to be solved by centralized Wiki documentation.

## ACRONYMS

CMMI	Capability Mature Model Integrated
KM	Knowledge Management
TS	Technical Solution (process)
ADD	Architectural Driven Design
ISO	International Organization for Standardization
SEI	Software Engineering Institute
SaaS	Software as a Service
TSP	Team Software Process
PSP	Personal Software Process
R & D	Research and Development
PM	Project Manager
LD	Lead Developer



## 1 Introduction

The Case Company in this thesis is a software company producing applications for Finnish industrial companies. Today these applications are more and more delivered as a service. Many software companies around the world use the CMMI model as a framework for defined processes and procedures. One significant part of the CMMI framework is the Technical Solution process. The purpose of Technical Solution (TS) is to select, design, and implement solutions according to requirements. This process is very significant for the development project to succeed since all the planning is included in this process. If Technical Solution is well designed, the project has all the keys to succeed within planned costs and schedule. If the solution is not planned right, the consequences can be severe. The Project might run into technical challenges causing the schedule to fail and losses in profitability. One challenge is to define how much effort the project should put into the Technical Solution planning. Too little effort causes plans to be too inaccurate and too much effort causes unnecessary costs. Therefore this thesis focuses on defining and creating a new software design process for the case company, which currently does not have a defined, suitable process for this area. Traditionally software design has been a similar process as planning almost in almost any engineering industry. However, software development is very different from more traditionally, product oriented engineering.

Designing systems, applications or products well and using past projects as references, prevents lots of problems in the production phase. Unplanned and undocumented design leads to the situation where individuals do not act as a team having a common goal. New components or technologies can be implemented by individuals without a broader view to technologies a company is using. Past projects have solved problems and have experiences, which can be utilized in new projects. Planning has been considered to be the backbone of every engineering product. Planning and documenting outcomes and experiences enables learning from past projects and spreading knowledge of experiences throughout the whole organization. It also enables also the identification of best and worst practices and therefore the improvement of the entire production. In software business, production processes have changed and varied over the years. Originally, software production had very similar phases and practices as manufacturing; clear planning, documenting and approval phases before activating the actual production resources. Nowadays, on the other hand, software production



and products are very different from manufactured goods or building unique products like ships; software is more intangible, abstract and not concrete. Therefore, there have been different approaches to planning and production practices mainly focusing on creating more agile and reasonable ways compared to the traditional manufacturing model. In manufacturing, planning must be absolutely finalized before mass production. If there is something missing or unclear in the plans, the whole production can go wrong. In software production, the product is easier to change and in most of the cases, the product changes during the development process. The process is more agile than in traditional manufacturing.

Many times, the outcome of the software project is misty for both customer and provider at the beginning of project. The Production of software is not capital intensive; almost anyone can buy a computer, the needed developing tools and can hire some space from cloud services to distribute Internet applications. The Internet has made it possible to distribute applications almost without expenses. Of course, at the same time the amount of software products offered has exploded and it is hard to differentiate between them. Therefore, planning and creating innovations during planning is important. Thanks to the introduction of cloud services and applications served over the Internet, technical documentation has become unnecessary from the customer or user's point of view. The Software service provider or developer is the only stakeholder needing technical documentation. Naturally, this change has an impact on the format, purpose, need and depth of technical documentation.

One important element for the successful company is the existence of good processes. Figure 1 below shows the Role of processes through three points; People that bring in skills and motivation, technology for tools and techniques to bring efficiency and the environment where the business is operating (Garcia et al. 2007:5).



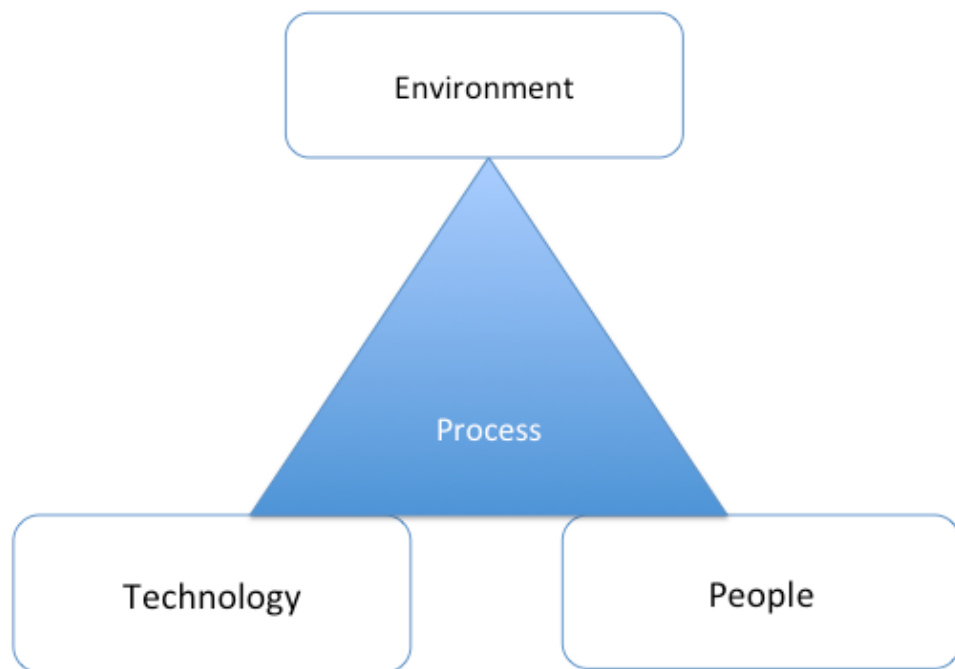


Figure 1. Holistic view of the role of process (Garcia et al. 2007:5)

This thesis intends to cover the improvement of a software design process of the case company, and especially the first version of the improved model for a process. All aspects shown in Figure 1 are to be considered; what is the situation in the case company (environment), what technologies are appropriate and what people designing software think about the themes process improvement should have. Today there are several concepts, methodologies or frameworks available that are designed for improving processes with many of them having a background in manufacturing and mass production. However, software products are very different when comparing to manufactured products; software products are abstract, logical components having no physical properties or relations to physical laws. Manufactured products are physical having physical attributes like mass, volume, color and so on. Therefore, software development processes have to have more specific and suitable models for process improvement still influenced by well-known frameworks. The case company has chosen to utilize the CMMI framework as the main guideline for process improvement. This thesis focuses on the CMMI framework but adopts best practices also from other frameworks used in software development.

The event of improving processes is always a change for a company and the people working there. There are two prerequisites for a new idea to take hold in an organiza-

tion: The idea must be proven to work operationally and people must understand why it works (Poppendieck et al. 2003). Therefore changes in processes have to be carefully planned, tested, communicated, guided and explained to every person involved in the change.

### 1.1 Case Company

The case company is a mid-sized software company providing currently both SaaS - products as well as tailor-made projects, often delivered as a service. The company's current strategy focus is to offer solutions for mobile work. This means solutions, which enable the right person to be in the right place, at the right time with all the information needed. The company has focused on industries such as Field Service, Maintenance, Transportation, Warehousing, Home care, Wood procurement and Inspections. Since 2007 the company has focused on the Software as a Service delivery model. The company has combined best practices from different frameworks, models and practices.

Some significant features to define the company include boundaryless communication environment and a high appreciation of colleagues and the atmosphere. This makes it possible to create process improvement projects based on ideas, visions and innovations of employees. Since the case company is a software company, the personnel consist of highly skilled, intelligent people, able to innovate and scrutinize the company from a development point of view. This kind of personnel tends to have good ideas to improve processes if someone is just willing to gather all the ideas, select the best ones and turn them into a new way of work.

The company was formed through a merger of three different but synergic companies, which had their separate, quite undefined processes at the time. The environment in these companies was partly chaotic; they operated with a heavy customer focus, with demand playing the main role when deciding which technologies to use. The deliveries were unique providing customer specific technologies and ways of work. Focusing on separate customers resulted in the creation of customer specific resources and teams. Since the processes or ways of work were not defined but instead were decided in connection with each customer, sharing resources over team boundaries was very difficult and the company was divided to customer specific areas, although serving customers very well. To refer to Figure 1 at the Introduction, people and their skills were



customer specific, the environment was scattered and the technologies were customer or even project specific.

In recent years, the company has improved its processes in the areas of Customer Management, Project Management, Financials, Human Resources and Engineering. The company has used the CMMI framework as a guideline in process improvement and has reached CMMI level 2 in an assessment made by K2 Quality Partners Oy. The objective of the company is to reach level 3 in the first quarter of the year 2013. One very important process area to be improved and defined is Technical Solution, meaning design and planning of systems and applications.

The Technical Solution process or software design practices are currently not clearly defined in the Case Company. There is no documentation or information available about design practices except document template for the Technical Plan. Different teams in the company design applications and systems differently. The Project Management process defines that there should be a Technical Plan created at the beginning of the project after the definitions. This is basically a document template describing what should be planned before the actual development project. It has quite a versatile structure constructed from real life experiences. It describes what should be planned when thinking mid-size or large software projects. Unfortunately, one size does not fit all; the template is too detailed for smaller projects and maybe too generic for larger projects.

In the Project Management process, there is an auditing point before implementation and the Technological Plan should be audited, commented on and checked. There are clear problems recognized with this approach. First, the template is in many cases thought to be too complex and the designers do not know which topics are important in each individual case and which are not. It is unclear to the users that should they fill in all the topics in the template and they don't know to what level of accuracy it should be done. An individual designer is unable to evaluate if a certain topic is relevant in this particular case and therefore tends to leave a blank. In that kind of a situation, it is also unclear what happens in the audition if something is left blank in the template. Secondly, the document template and versioning is thought to be too troublesome.

It is also unclear whether a new version should be created when the document is updated. Designers do not understand clearly the purpose and target stakeholder group



of the document. The designers are also wondering should the document be written using terms that are familiar for specialists or terms familiar for customers.

Creating this kind of a document is felt to be mandatory without any clear value. All this often leads to the situation where the Technical plan document is left undone and some of the relevant information is written to the company wiki but without guidance, encouragement or acceptance.

The current Technical Solution or software design practices create the following problems: The projects are probably not sufficiently well planned and this causes disturbances especially in the situations where new technologies are to be implemented in the current project. In these situations, the best approach could be to conduct a research study for the new technology, framework or module. Unfortunately, when the project faces such a situation, it is usually too late to start the research, which leads to a problematic situation; whether to conduct the study within a separate stream or include the study to the start of the project.

It is unknown if it is efficient to make detailed plans and if it will make the company to be safe enough considering the risks. It might be more efficient just to plan with the focus on more universal principles to guide the flow of development towards the best possible end. Re-usability is at a low level, which means that when a new project starts, there is no 'mindset' to search benefits from previous projects or experiences. It is even considered a normal procedure to start every project from a clear table, which gives ultimate freedom for the project to succeed technologically, but not necessarily economically. In a sense, this kind of approach is the most beneficial considering customer unique needs. This approach also ensures that every project can use the most recent technology.

One problem is that knowledge and best practices are not spread throughout the organization. Information regarding technical solutions is either in separate planning documents or in the company Wiki. If the approach is to have information only in planning documents, these are not updated and only an overall plan at the start of the project is created. When actual production is not documented it causes a situation where personal expertise and experiences are not spread. This causes troubles with resourcing; certain people are skilled to use a certain technology and are eventually overloaded with work. If the company could widen the expertise and spread knowledge, it could clearly ease up any resourcing problems.



When trying to find an approach for the solution at the bidding phase, there is no easy way to find similar reference projects. This leads to a situation where the planner must call several people and ask if they have any experiences of similar cases. If the planner is not active at this phase or too busy to ask from numerous people, the approach is very narrow and only based on the experiences of one single planner.

As a summary, the company has a document centric approach to the Technical Solutions and planning. This causes a situation, which does not enable efficient knowledge sharing and learning, which should be one of the objectives in process improvement. Knowledge sharing would ease the design phase since similar reference cases could be utilized.

## 1.2 Business Problem and Objective

The overall business problem is that the company does not have a proper planning and design process that would be well documented and mature enough. Therefore the objective of this study is to create a suitable Technical Solution process for the company to harmonize the planning and implementation practices. User experiences need to be documented and shared and knowledge transferred by making documentation easier to share. The new process should also enable a learning organization, which refers to the overall objective of the case company to enable employees to learn through other person's experiences in different projects.

A Technical Solution process describes the methods used to design quality solutions. Typically the solution consists of planning solution choices, the design and implementations. This thesis focuses on planning and design. The solution is considered to be either a tailor made system, application for a customer or a product developed inside the Company. The solution can also be a technical design of a software component or a module. The main objectives of the Technical Solution or planning are: Most reasonable solution (logical solution), cost-effective solution (creates competitive advantage), re-usability of utilization of previous solutions, solution which is in line with the company wide technology decisions and suitable to customer's needs or wishes.

Process improvement is a functional, operative, continuous organizational project which always aims to reach some organizational business objectives. Based on Ahern et al. there are multiple objectives. The first of the seven common objectives mentioned



by Ahern et al. is producing quality products or services for the customers. The second is creating value for the stockholders. Being an employer of choice for the employee is the third objective. The fourth is enhancing customer satisfaction to maximize customer loyalty and increasing market share is the fifth objective. The sixth objective is implementation of cost savings and successful practices. The last of the seven objectives is gaining an industry-wide recognition for excellence (Ahern et al. 2008:56). In a process improvement project, it might be a good approach to first define the business objectives and benefits the project should achieve. It could also be a good idea to prioritize objectives and use this to steer the improvement projects.

## 2 Research Approach

This section overviews the research methodology used in this thesis. The section also presents the research process including the research conducted on best practices in the industry and the data collection process. It also discusses how the reliability and validity of this thesis will be secured.

### 2.1 Research Methods and Process

Two common research strategies are quantitative and qualitative research methods. The one selected for this research is the qualitative method. The qualitative method provides a depth and richness of data that allows close involvement from the researcher. Quantitative research is an objective type of scientific inquiry where the researcher tries to be detached from the actual subject of the study. Quantitative is a more measurable and case oriented research method (Tomal 2010:3). The qualitative methods in this study are the interviews used to collect the data.

The selected overall research approach in this Thesis is Action Research (AR). In action research, data is not only collected and analyzed as in other research approaches but also implemented in actions (Tomal 2010:10). Action research is a systematic process of solving problems and making improvements where the researcher functions at the same time as a researcher and an agent in the organization. Action reflects the fact that research is done for the process in action. Kurt Lewin, a social psychologist and educator first introduced the method in the 1940's. He presented that research work should not separate investigation from the action needed to solve the problem. The process he presented was cyclical having a non-linear pattern of planning, acting, observing and reflecting on the changes. The first phase is to identify the problem area, and determine the objective. Selecting a meaningful question, limiting it and planning carefully are important in this phase. The second is to collect data and organize it. Data collection can be made from several sources such as interviews, journals, photos, memos, questionnaires, focus groups, audio tapes, diaries, field notes, surveys and logs of meetings. It is important to select the data, which is most appropriate for the issue being researched. The third phase is to interpret and analyze the data. It can be best to analyze and identify major themes. Data that is not quantifiable or statistical can be reviewed having a holistic approach and only the most important elements can be noted. The fourth phase is to plan and take actions based on the data and interpretations, an important part is to study and observe the change caused by actions. Data



about performance is good to collect for later interpretations. The last phase is to evaluate the results and measure improvement if possible (Ferrance 2000:1-9).

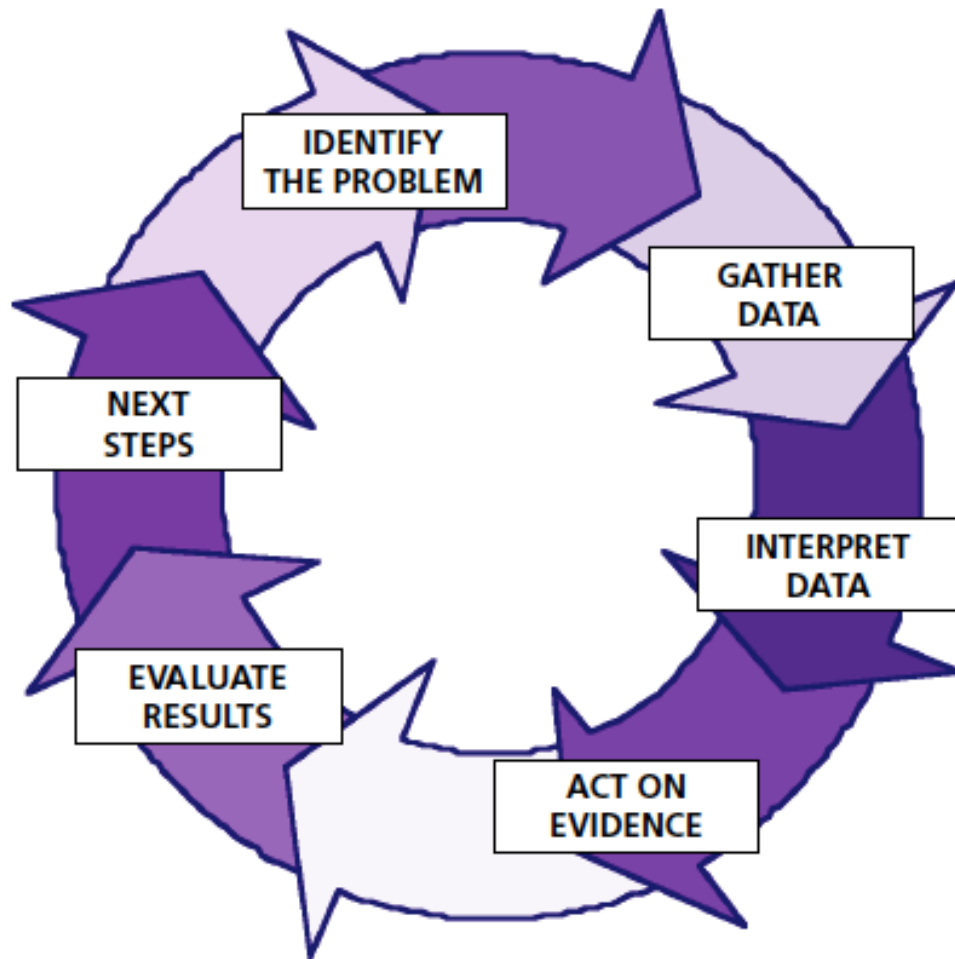


Figure 2. Lewin's Action Research cycle (Ferrance 2000:9)

In this thesis the research process is a circle following the principles of action research. First the business problem is identified and current state is analyzed and documented. The current state analysis starts with the company wiki search. After that, the current document templates are analyzed. The company Tech Steering Group then discusses the current situation. The next phase is to seek literature and other such sources for theoretical background information. Books, studies and articles are the best sources to find industry best practices. Literature data is going to be collected by searching appropriate articles from several databases, using Metropolia University of Applied Science's portal for database connections. Another source planned for literature and articles is to find books covering the topic from Metropolia's library and the case company's library. At first, the amount of articles and books are going to be collected and browsed

through and the most relevant are going to be selected to be the actual sources and references for this thesis.

Phase three is to collect data from the people working for the company. In this Thesis the plan is to have a companywide inquiry and targeted interviews to collect the data. Phase four is to form the first version of the new model by combining best practices from literature and the data. After careful creation of the model, the model is tested in the pilot projects. Feedback will be collected from the experiences in the pilot projects and it will be used to improve the model.

The research process designed in this Thesis is illustrated in Figure 3.

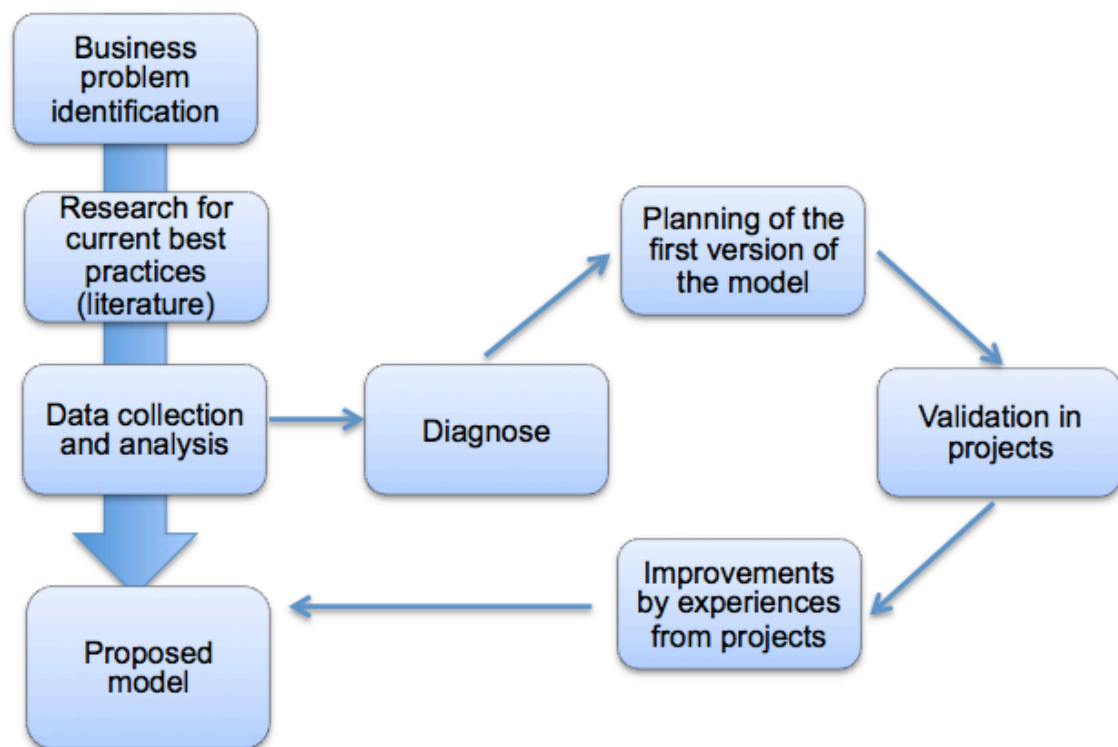


Figure 3. Research process designed for this thesis.

As seen in Figure 3, the process starts with Business problem identification. The second phase is to research for current best practices, mainly searching sources from literature and articles. The third phase is the data collection, which leads to diagnosing the data. After that, the first version of the model can be created and validated in the piloting projects. The validation should cause some improvements to the first version of the model.

## 2.2 Data Collection and Analysis Methods

In this study, the plan for the data collection is divided in two, separated phases; one overall inquiry and several, targeted interviews concentrating on getting ideas of the optimal design process from the very people designing software. The collected data will be discussed in sessions kept in the case company's Technology Steering Group (TSG), which is a team discussing and making decisions about selected technologies to be used in the case company's projects.

The third phase in the research process is the literature search and familiarization with the topic through articles, books and other publications. The first objective in this phase is to gain enough knowledge to be able to formulate correct and reasonable questions for the data collection.

Phase four in the research design is to select suitable questions to be asked in an overall inquiry made available for the whole personnel of the case company. The questions will be selected based on best practice content. The overall inquiry is going to be considered as collecting method to reach each and every developer interested in influencing the creation of the new process.

Another relevant data collection method is going to be the targeted interviews among lead developers. Lead developers are typically improvement-oriented people having visions of the future target state. It is expected that the best input will be gathered from these interviews. Questions in these interviews will be constructed based on the findings from literature, and reflected on best practises found. Still, these questions should be open questions enabling a free flow of discussion and drill down questions. The results of the overall inquiry can also influence to the decision of questions and topics.

Data collected from both literature studies and interviews will be used as a basis for a conceptual framework. The conceptual framework will be discussed in the TSG meetings before formulating the first version on the model. The first version of the model will be tested in the pilot projects to get input for the improvement cycle.

The details of the data collection are specified in Table 1.



Data from	Participants	Topics	Documents
Inquiry (5.-9.11.2012)	The case company whole personnel, 50 persons	Selected questions based on literature research best prac- tises.	Inquiry will be pub- lished in Digium service the case Company is using.
Interviews (17.-21.12.2012)	Lead developers and Technical architects (5- 6 persons)	Selected, open questions based on literature research best practises and results of the in- quiry.	Field notes
Tech Steering Group session (6.11.2012)	Members of Tech Steering Group (5 per- sons)	Selected, open dis- cussion topics based on literature research best prac- tises.	Notes. Wiki Blog written.

Table 1. Details of the data collection.

As seen in Table 1, the inquiry was planned to cover the whole organization. Interviews were planned to have a set of Lead developers who are the very people designing the software. After the inquiry and interviews, it was planned that the Tech Steering Group would discuss the results and analysis of the data collection.

The inquiry (line 1 in Table 1) was targeted to the whole personnel of the case company knowing that not all roles in the company are related to Technical Solutions. At the time the inquiry was published, there were 50 employees in the case company. The inquiry had 14 selected, relevant questions. The inquiry was planned to have both qualitative and quantitative questions. The role of the quantitative questions was to warm up the respondents so that they would be tuned to answer the qualitative, more relevant questions. The quantitative questions were simple, not leading questions like “Do you think that the process should be improved in this company?” This quantitative data gathered is not used in this thesis. Still, the main method of the questionnaire can be considered to be qualitative, since the relevant questions allowed the respondents to write down their ideas and visions to improve the Technical Solution process. The inquiry acted also as a wakeup call for the company employees informing them that the

process is going to change and their opinions are both important and more than welcome at every stage of the development.

The inquiry ended up having seven respondents and the answering time was one week. Several reminders were sent to the whole personnel to answer this inquiry. The inquiry was designed to have both multiple choice questions to guide the thoughts of the respondents to the right direction, and open questions to give the respondents the possibility to describe their opinions. The main focus of the inquiry was to have open questions, freely written comments and ideas from the respondent group. People answering the inquiry were able to write quite deep answers to the questions that allowed free writing. The results of the open questions from the inquiry were also used in one-to-one interviews to get as deep comments and ideas as possible.

The next method in the data collection was targeted interviews with Lead developers and Technical architects; the very people related to the Technical Solution process. The interviews were considered to be a method of qualitative data collection. The interview questions were connected to best practices found from literature, especially the CMMI framework.

The gathered data was analyzed in a workshop with the Tech Steering Group (TSG) team, which has the authority to make decisions considering technologies the company is using. In the workshop, the members of TSG added their own ideas or suggestions to the data and gave comments about the gathered data. As a result of this workshop, the first sketch of the prototype was formulated.

### 2.3 Reliability and Validity

Measuring the results of a research should be done through considering *Validity* and *Reliability*. This study is a qualitative research, therefore validity and reliability are defined based on qualitative research methods. In qualitative research, the researcher is an instrument; interviews and observations are dominant tools. A good qualitative study helps to understand a situation, which would otherwise be confusing. The study helps to generate an understanding in some matter. In this sense, it is hard to determine the reliability of qualitative study. In qualitative paradigms the terms credibility, neutrality, confirmability, consistency, dependability and applicability are essential criteria for quality (Golafshani 2003:601). The terms presented can be thought of as aspects of “trustworthiness” of the study.



A qualitative method can have certain kind of evaluations and standards, even auditors. Morse et al. (2002) found that rigor does not rely on special procedures external to the research process itself. The researcher should maybe just be competent and familiar with qualitative research methods and perform a wide enough data collection procedure. "In summary, we need to refocus our agenda for ensuring rigor and place responsibility with the investigator rather than external judges of the completed product." they argued.

Validity measures whether the outcome of the study responds to the objective set for the study. Validity also relates to how the data has been collected to help the research question to be answered. In this study a survey will be used for all of the staff and an interview round will be organized to secure the developers' perspective. Validity also requires the correctness and credibility of the description, conclusion, explanation and interpretation. By using quotations of the informants efforts will be made to give a credible interpretation.

Validity in this study is secured by the fact that the case company does not have a proper process for the software design. Therefore, a new process will be useful for the company. Since all software has to be designed before development, the process is very important to be able to successfully develop competitive software applications. The Company management has defined Technology Solutions as one of the processes to be improved in 2013.

Reliability includes how error free, trustworthy and replicable the research is. Reliability in this study is secured by a careful background study and interviews carried out inside the company. The interviews are going to be conducted among Lead developers, who are the people with career background from several different software companies. These people have seen both best and worst practices during their career. The interview transcripts will be checked by the interviewees for better reliability. The interview questions will be attached as the appendix. This study has done its best to ensure the reliability of the study.

This section presented the research approach and the methods that will be used to collect data and find best practices from literature. The reliability and validity of the study will be evaluated at the end of section 7 Discussion and Conclusions.



### 3 Best Practice for Process Management

This section discusses different models for process improvement used in the software industry. The section is divided to seven sub-sections. The first is an introduction to definitions of frameworks, principles and practices. The second section describes the most relevant framework, i.e. CMMI for Development and the Technical Solution process model of it. The third sub-section is about Lean methods, especially how these methods are used in the software industry. The fourth sub-section introduces Agile methods together with Architecture Driven Design. The fifth sub-section describes the Six Sigma process model, again as used in the software industry. The sixth sub-section presents the objective of Learning Organization and knowledge sharing which is common to almost all frameworks. Knowledge sharing is also defined to be a very important efficiency and competition factor for modern software companies. The seventh sub-section wraps up the findings from the literature and discusses best practices usable for the conceptual model of the Technical Solution process.

In engineering, many different models have been created to describe possible approaches to improve processes and ways of work over the years. These approaches are generally called as Model based approaches. Model based process improvement involves the use of a structured framework to guide the improvement of the organization. Process improvement is typically aimed to increase the capability of work processes. Capability is considered to be the ability to produce planned results with the resources allocated and timeline set. When capability increases, the process becomes more predictable and measurable. In other words, the process becomes more reliable and comparable to the objectives set. When processes are capable enough, the most important reasons for poor quality and productivity are eliminated or at least controlled (Ahern et al. 2008).

There are several frameworks and even standards widely used in Technical Solutions and software design processes. Many of these frameworks define very common design patterns used in several fields of engineering. Still, there are specialized versions of frameworks designed for software development.

*Framework* is defined to be a structure for supporting or enclosing something constructive, especially a skeletal support used as the basis for something being constructed ([www.thefreedictionary.com](http://www.thefreedictionary.com)). *Principles* are guiding ideas and insights about discipline



while *practices* are what you actually do to carry out principles. Principles are universal, but not always easy to see. Practices give specific guidance on what to do and they must be adapted to the domain (Poppendieck et al. 2003).

### 3.1 CMMI for Development

The first approach the case company utilizes for process development is CMMI for Development. CMMI (Capability Maturity Model Integration) is a process improvement approach created by Software Engineering Institute (SEI) at Carnegie Mellon University U.S.A. It has been created by a group of experts from the fields of industry, government and university. According to SEI, CMMI helps "integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes." CMMI focuses on improving processes in an organization. The model contains essential elements of effective processes and describes an evolutionary improvement path from ad hoc, non-managed, immature processes to disciplined, mature processes with improved quality and effectiveness. CMMI is a framework, which emphasizes integration to company's existing processes, possibly, improving them.

CMMI is considered to be a framework for coordinating process improvement efforts and measuring and monitoring the status of those efforts. CMMI provides a basis for benchmarking the capability of individual processes and maturity of the company. The scale is 0-5 where 5 is the best. Level zero is considered to be the starting point of process improvement in company having no defined processes at all.(Ahern et al. 2008:36). Maturity levels are related to organization's overall process improvement in multiple process areas. Capability levels are more focused on describing an organization's achievements in individual, organization specific process areas.

The levels of the CMMI framework are presented in Table 2.



<i>Level</i>	<i>Continuous Representation Capability Levels</i>	<i>Staged Representation Maturity Levels</i>
<b>Level 0</b>	<b>Incomplete</b>	
<b>Level 1</b>	<b>Performed</b>	<b>Initial</b>
<b>Level 2</b>	<b>Managed</b>	<b>Managed</b>
<b>Level 3</b>	<b>Defined</b>	<b>Defined</b>
<b>Level 4</b>		<b>Quantitatively Managed</b>
<b>Level 5</b>		<b>Optimizing</b>

Table 2. Capability (0-3) and Maturity (4-5) levels in CMMI (Software Engineering Institute 2010:23).

As seen in Table 2, capability and maturity have different levels. Both share the levels from 1 to 3. After that, the company is considered to improve its maturity by measured and optimized processes.

An important objective for many companies is to achieve level 3, Defined. It means that processes are well characterized, documented and understood. There are procedures and tools used similarly throughout the company. The organization has set so called standardized procedures even though the procedures can be tailored within agreed limits. At level 3, a defined process has clear definitions of the purpose, inputs, entry criteria, activities, roles, measures, verifications, outputs and exit criteria. At level 3, the company further develops its processes towards the practices where processes are measured using statistic or quantitative techniques for further improvement (Software Engineering Institute 2010). In order for an organization to be officially 'certified' for CMMI, and have their rating published, an external certified team or company of appraisers must conduct a review of the evidence that would support the CMMI Practices and Goals required for the given level rating. Typically this is a major undertaking with a small team or specialized company of external auditors collecting information using interviews, reading documentation etc. and then analyzing this information to produce an appropriate, objective rating.

One guideline is to use common sense when interpreting the model for the company. In CMMI, improvement should be driven by business value; the focus should be where most value is gained by improvement. Therefore, CMMI does not require immediate focus on all process areas. It is recommended that the company implement the practices that obviously make sense for the organization (Garcia et al. 2007:58). When incrementally improving in processes, maturity raises to a higher level.

There is a version of the framework called CMMI for Development (CMMI-DEV), which contains practices covering project management, process management, systems engineering, hardware engineering, software engineering and other supporting processes used in development and maintenance. CMMI-DEV suits very well for software companies. It is therefore a reasonable choice for the case company too. The framework has a dimensional structure containing the first 16 different core process areas, 1 shared process area and 5 development specific process areas.

The Software Engineering Institute has defined 22 process areas listed below in alphabetical order by acronym:

- Causal Analysis and Resolution (CAR)
- Configuration Management (CM)
- Decision Analysis and Resolution (DAR)
- Integrated Project Management (IPM)
- Measurement and Analysis (MA)
- Organizational Process Definition (OPD)
- Organizational Process Focus (OPF)
- Organizational Performance Management (OPM)
- Organizational Process Performance (OPP)
- Organizational Training (OT)
- Product Integration (PI)
- Project Monitoring and Control (PMC)
- Project Planning (PP)
- Process and Product Quality Assurance (PPQA)
- Quantitative Project Management (QPM)
- Requirements Development (RD)
- Requirements Management (REQM)
- Risk Management (RSKM)
- Supplier Agreement Management (SAM)



- Technical Solution (TS)
- Validation (VAL)
- Verification (VER)

The process areas in CMMI include both Generic Goals (applicable to multiple process areas) and Specific Goals. Both goals have practices, which can have then subpractices. The process area is defined by having a purpose statement, introduction and relations to other process areas. The model components are grouped into three categories: required, expected and informative. The square-shaped required components are important to achieving process improvement. Achievement has to be visibly implemented in the organization's processes. The diamond-shaped expected components describe the activities important to achieving the component whereas the oval-shaped informative components help users to understand the required and expected components. The informative components can be e.g. notes, examples or references.

Figure 4 illustrated relationship between goals and practices.

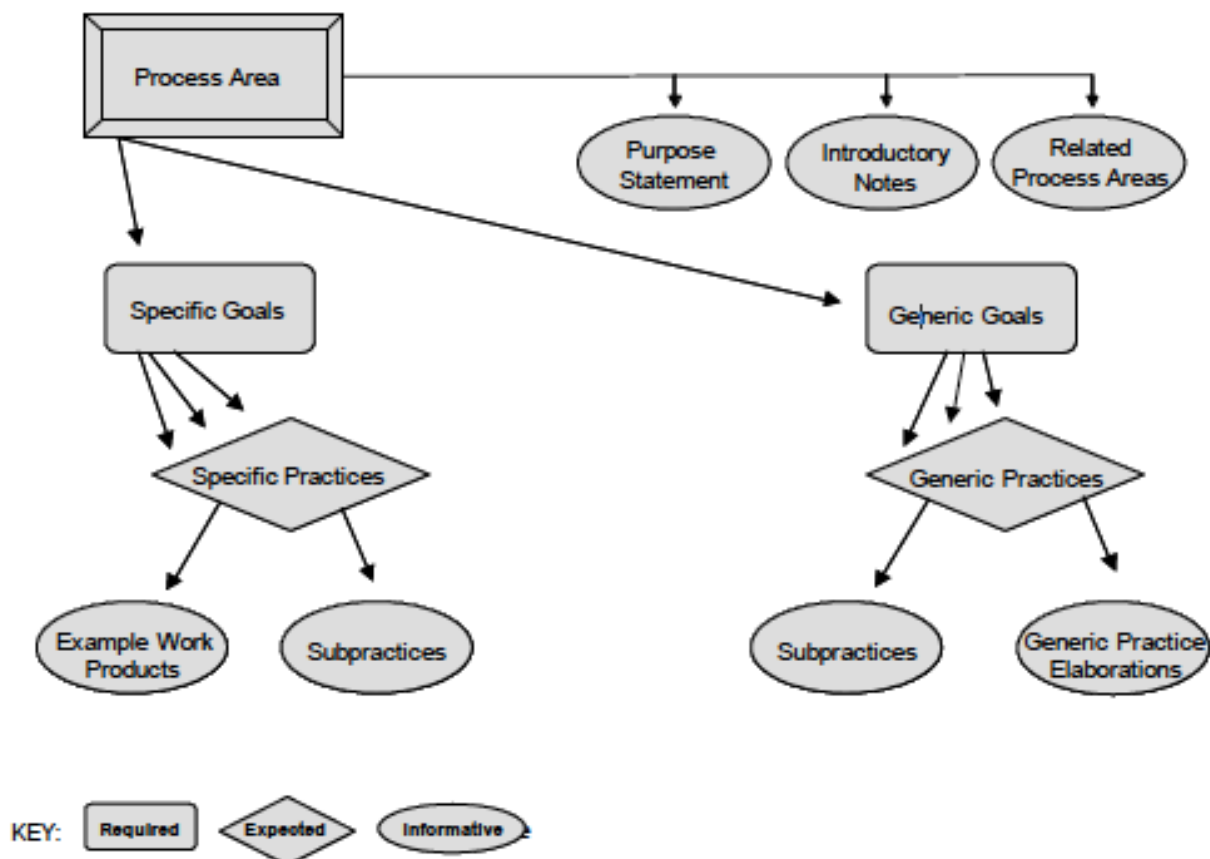


Figure 4. CMMI model components (Software Engineering Institute 2010:10).

One of the process areas is the Technical Solution (TS), which is the focus of this Thesis. The TS process area's purpose is to select, design and implement solutions to fulfill requirements. This process area is related to the process areas of Requirements Development, Verification, Decision Analysis and Organizational performance management. The Specific Goals and Specific Practices of the TS process area are:

- SG 1 Select Product Component Solutions
  - SP 1.1 Develop Alternative Solutions and Selection Criteria
  - SP 1.2 Select Product Component Solutions
- SG 2 Develop the Design
  - SP 2.1 Design the Product or Product Component
  - SP 2.2 Establish a Technical Data Package
  - SP 2.3 Design Interfaces Using Criteria
  - SP 2.4 Perform Make, Buy, or Reuse Analyses
- SG 3 Implement the Product Design
  - SP 3.1 Implement the Design
  - SP 3.2 Develop Product Support Documentation

Similar international standards are ISO/IEC 12207, standard on software life cycle processes, ISO/IEC 15288 international standard, on system life cycle. Six sigma and Lean engineering are also considered to be similar frameworks (Garcia et al. 2007:22).

Thus CMMI is very relevant for the Technology Solution process development since it has this process quite well defined.

### 3.2 TSP and PSP

The second and the third framework essential to the case company are TSP (Team Software Process) and PSP (Personal Software Process). They are presented together because the PSP-training program has different levels and finally leads to TSP. Accordingly, personal process improvement gives readiness to operate in a team. Both are operational frameworks, which help teams having managers and engineers to organize projects and implement software projects producing either tailor made applications or products. The focus of TSP is to improve quality and productivity with different stakeholder needs, typically cost and schedule. PSP focuses on helping engineers to improve performance by using discipline procedures. Watts Humphrey, an American software engineer, developed TSP and PSP. He is considered to be the father of software quality. TSP and PSP are applied to the more complex process improvement model CMMI.



PSP and TSP focuses to provide experts with skills to negotiate with the management when the management requires project work that a too tight schedule. One guideline is to make a thorough plan and defend it. Costs and schedule are defined to be a requiring force to make plans. Without a plan, the team is generally in such a rush to code and test that developers cut corners and don't do as good a job as they could. Teams and managers should establish goals, define a role for each team member and define risks, as well. The teams should also estimate the efforts and have at least weekly meetings to follow up on progress made. A suggestion is to divide the projects to development cycles. It is important to follow the planned and actual efforts to be able to report on the status. Each development cycle should have a Post Mortem review to be aware of the progress and gather lessons learned.

The focus of PSP is on individual skills such as improving estimating and planning skills, communicating commitments the developers can keep and reducing defects in daily work. The main target in PSP is to help developers to produce quality products on schedule. One most centric principle in PSP is to measure performance and to use past measurements to analyze the trend of the performance improvement. PSP has four core measures, i.e. the size of the product part, time required to complete task, number of defects in the product part and success with the schedule.

The mindset behind TSP and PSP is that when plans are approved to be good enough by the team, the team can be committed to produce such a product part. The principle that there is a separate team plan and a personal plan is supporting the ability to be committed to results.

Figure 5 presents the structure in TSP and PSP.



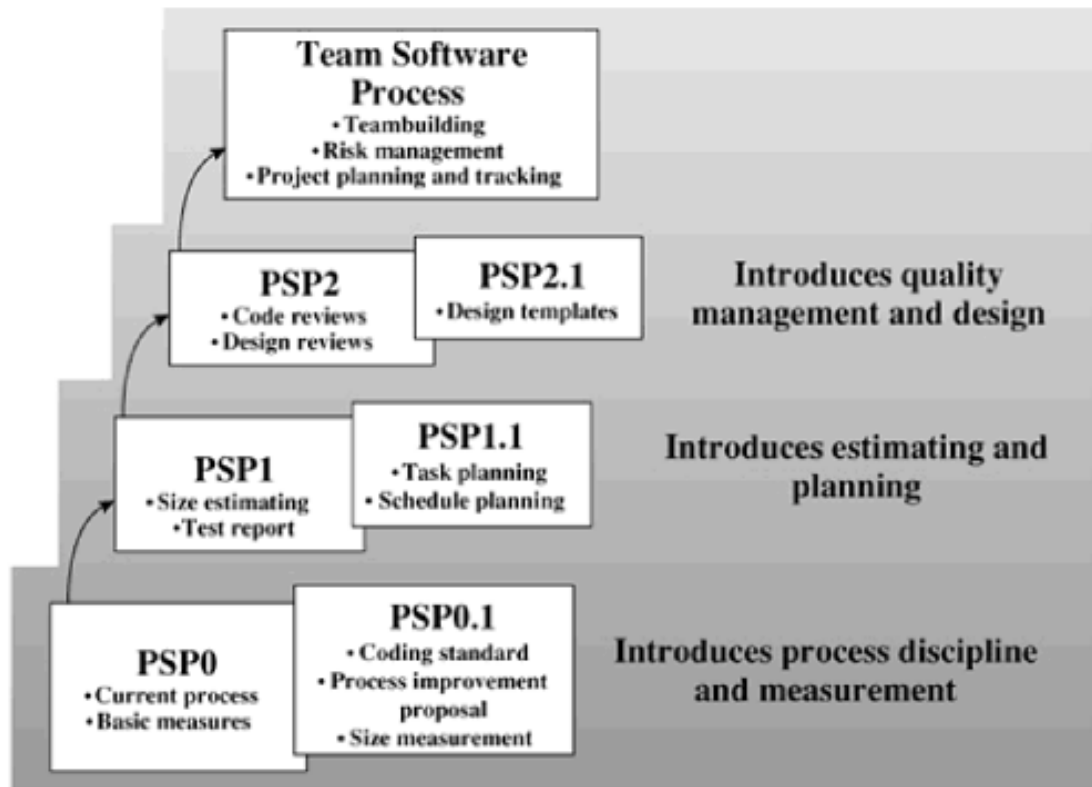


Figure 5. Structure in PSP and TSP (Humphrey 2005:7)

Figure 5 shows that the way to achieve Team Software Process is to improve Personal Software Processes first. The first level (level 0) concentrates on having coding standards and measurements. The second level (level 1) concentrates on estimating workloads and planning schedules based on estimates. The third level (level 2) introduces quality management and design templates to improve quality and efficiency. Improvement at personal level enable improvement in team level by planning, risk management and team's motivation to succeed as a team.

### 3.3 Lean Methods

The fourth framework essential for the case company is Lean. Lean methods and Lean thinking originates from late 1940's when the company named Toyota was facing a situation where customers did not have so much money because of the World War Two and thus cars had to be cheap. The Japanese market was small and the cheapest way to produce cars was by mass production. Toyota's challenge was how to produce small quantities of models but keep production inexpensive. They created the Toyota Production System, which emerged from a whole new way of thinking about manufacturing, logistics and product development. The essential vision is that there is always a

better way to provide products (Ahern et al. 2008). There are seven main principles in Lean thinking. The first and maybe the most dominating is to eliminate waste. Waste is considered to be anything that does not add value to a product. In software engineering, for example, required documentation nobody uses is considered to be waste. The second principle is to amplify learning. Software technology is developing fast so learning and knowledge sharing is greatly emphasized. One of the most valuable learning factors is feedback from the process. In software development it means practices such as running tests as soon as code is written, trying ideas by coding instead of planning in detail and showing potential user screens to the customer as soon as possible. The third principle is to decide as late as possible. That can be compared to the situation in software development where definitions have to be ready before you can truly make a final decision concerning a technical approach and components used. The fourth principle is to deliver as fast as possible. In developing software, speed is very important because the final product is often quite abstract when development starts. Therefore it is important to have feedback from the customer iteratively as soon as possible. The fifth principle is to empower the team. Involving developers in detailed technical decisions is fundamental to achieving the best possible result. The sixth principle is to build integrity in. Software with integrity has coherent architecture and usability is maintainable, adaptable and extensible. The final principle is to see the system as a whole. Software development teams are typically quite small and consist of different experts. Experts tend to think that their field is the most relevant and important in the final delivery. Seeing the importance of all parts is needed to achieve a common goal (Poppendieck et al 2003).

Since removing waste is a dominating principle in Lean, it is divided to seven different types: Inventory, extra processing, overproduction, transportation, waiting, motion and defects. Learning to find and see waste is a constant process changing the way an organization thinks what is truly important. As mentioned, paperwork or documentation that no one reads adds no value especially in software development. Documentation should be kept as simple as possible. Even if someone truly uses documentation, there should be constant search for the most efficient, effective means to transmit information. The problem is that the documentation cannot include all the information the next person in the production process needs. There is always some part of the information, which has to be exchanged verbally, presented and explained.



Lean methods cover also engineering practices. The Key Lean Engineering practices are: Identify and optimize enterprise flow, Implement integrated process development, ensure seamless information flow, ensure process capability and maturation, optimize utilization of people, develop trust, commitment and accountability, develop learning environment, make decisions at lowest possible level, always focus on customers and understand 'value' from a view of a customer (Ahern et al. 2008)

### 3.4 Agile Methods and Architecture Driven Design

The fifth and sixth relevant frameworks for the case company are Agile and Architecture Driven Design (ADD). Agile methods are focused on minimizing architectural design whereas ADD focuses on more detailed architectural design.

These frameworks are presented together because from this study's point of view, they both deal with Architectural design although quite differently. Rapid business changes need rapid software development methods. Agile development methods are based on an iterative and incremental process. The most well-known agile method is probably Scrum. In Scrum, a project is divided to Sprints, which are time-boxed plans, which contain a system's features in a prioritized order. Sprints are iterative when compared to each other and there are planning sessions and reviews between Sprints. Shared information is in a special role and the model includes daily meetings concentrating on what was done, what is going to be done and whether there are any problems. While more traditional development and engineering processes emphasize clear process phases like planning before production, in agile thinking it is accepted that requirements and solutions evolve during the project. The methods also emphasize collaboration between the customer and self-organizing development team (Holcomb 2009:1-4).

In agile methods planning and design are adaptive and changes are made rapidly. Therefore, very detailed or careful planning is not necessarily reasonable although Architectural fundamentals must be decided before the actual development or implementation of the application. If the project has a very exact schedule, the application features should bend. Agile methods do not fit in fixed priced projects unless the customer accepts that lesser features are delivered if changes occur during the project. Agile methods are a very efficient way to do software projects when the customer does not truly know their needs, which is often the situation in software projects (West 2009:1). Still, the use of agile methods requires deep customer trust against vendor and commitment



to co-operative, collaborative development model. A tight budget and a requirement for a fixed price and fixed content is not necessarily the right environment for Agile methods. Instead, bending requirements within the budget is the right way to start thinking about using Agile.

There are also mixed ways to work where Agile production methods are combined with Architecture Driven Design (ADD) method. The ADD method focuses on something agile developers often ignore; the overall system structure that the quality attributes shape. ADD differs from agile methods' core practices because it emphasizes quality attribute requirements explicitly using architectural tactics. The quality attributes shape the architecture's structure, and functionality is allocated to that structure (Nord et al. 2006:4). Architecture is documented with diagrams or 'views' rather than writing detailed documents. The views are selected by stakeholders' needs; a customer needs a simplified view when on the other hand a software designer needs a more detailed, technology-oriented view. The views guide development even in a situation of changes and allow the possibility to make detailed decisions when the time is right.

Figure 6 illustrates the development model in Agile methods.

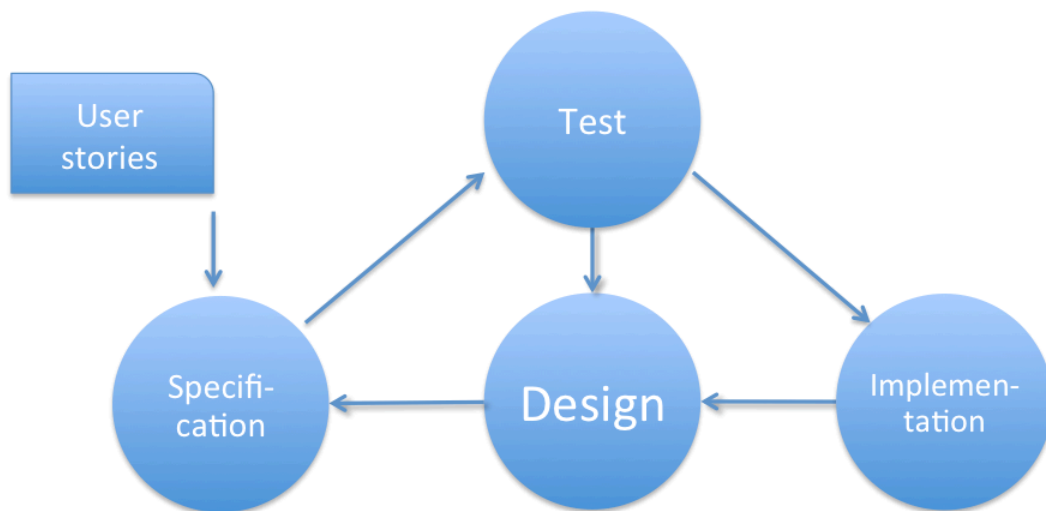


Figure 6. Development model in one of the Agile methods: Extreme programming (Nord et al. 2006:3).

In Figure 6, the arrows describe the direction of information flow; the end point of an arrow pulls information from the starting point. The first phase in Agile projects is the Specification phase. User stories collected from the customer by using interviews are influencing the specifications. The design connects to the business goals (specifica-

tions) and quality attributes (testing) together with experiences in implementation (development) give information to change the design if needed. In this approach, design is flexible to the changes during the projects. Agility could be one of the focus areas in the data collection phase.

Figure 7 presents the development model in Architecture Driven Design.

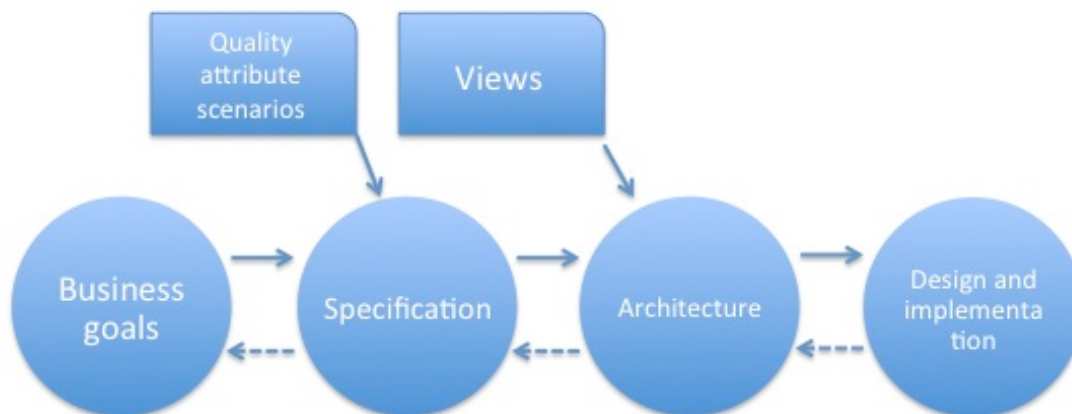


Figure 7. Development model in Architecture centric model. (Nord et al. 2006:3).

In the Figure 7, the dashed arrows describe the evaluation of the start point whereas the solid arrows describe the synthesis of the end point from the start point. In this model, business goals should generate quality attributes as a result of quality attribute workshop. These attributes should influence the specifications. Views are considered to be documentation based on different diagrams. Design and implementation is done based on the architecture (architectural conformance). When designing architecture, specifications are considered from the perspective of how the architecture would be the most cost-effective solution. The solution should not have any major tradeoffs from the point of view of future development or software maintenance.

When combining agile methods and ADD, the approach is that software architecture becomes clearer and more detailed by incremental steps in the project. The level of detail is flexible.

The aim of Agile architecture is to be bendable. Bendable means that only major elements of the architecture are planned and the architecture then allows more detailed planning when needed. Developers do just enough architectural planning to ensure

that the design will produce a system, which will fulfill quality demands and attributes. The first iteration plays a crucial role in defining overall structure.

### 3.5 Six Sigma

The seventh framework, which could influence the Technology Solutions process, is Six Sigma. Six Sigma efforts originated at Motorola Inc. 1986 when the company had problems related to quality. They created a set of processes and strategies to improve in business and achieved an 80% reduction in costs, improved quality and doubled their productivity. Six Sigma seeks to improve quality of process by identifying and removing errors through quality management methods. Measuring is a basis to find out variance in errors. It is a management program where cultural change and being proactive instead of reactive is very important. Since Six Sigma is quality centric, it requires understanding of the cultural aspects of quality. It also required deep management commitment and strategy change to highlight the change of culture and mindset. Six Sigma is also about open communication; facts, challenges and successes should be communicated transparently. Frequent communication in all levels helps to find out reasons for defects, improve the process as quickly as possible and increase customer satisfaction through improved quality (Rupa 2011:1-8).

Six Sigma is meaningful for software production because of the focus on quality. Software quality is often seen as a very complex topic and it is maybe one of the most ignored topics in the whole industry. Many companies have seen that commitment to quality speeds up development, reduces costs and allows new features to be added more easily. It is estimated that finding and fixing a software problem after delivery costs 100 times more than finding and fixing it during the design phase. Six Sigma for software is the application of the framework to the whole software development lifecycle. It is about measuring, analyzing, reducing defects, optimizing cycle time, schedule slippage and considering different variations in processes. Since software does not have physical attributes, conventional manufacturing metrics cannot be used, which makes software development processes difficult to measure. For software development, Six Sigma is more like a model to achieve continual process improvement just as the CMMI model is. Six Sigma focus is on quality planning, requirements gathering, design, testing and maintenance (Rupa 2005:2-3).

Software professionals tend to prefer CMMI instead of Six Sigma although some principles of Six Sigma are mixed with processes in different companies. There are suc-



cess stories, which show better success rates, reduction of schedule variation and increased customer satisfaction when Six Sigma principles are used (Rupa 2011:4). Six Sigma's analytic focus might be too heavy for most of the software companies. It is possible that agile methods give a better answer to software development challenges in a rapidly changing and evolving business. There are also segments in the industry where absolute quality is needed such as for example nuclear power plant systems. For these kind of system development projects Six Sigma's quality focus is more appropriate than for example consumer applications. Compared to CMMI, Six Sigma could provide good tools for strong, quantitative measuring, which is used at CMMI levels 4 or 5. The principle of high quality is very recommendable since customer satisfaction depends essentially on the number of defects in software.

### 3.6 Objective of Knowledge Sharing

Knowledge sharing and creating atmosphere and environment of learning organization is considered to be one of the most important success factors in modern software business, since, as we saw in our initial Figure 1, technology and environment are linked with the actions of people. Demands in today's highly competed software industry such as short lead-time, frequent introducing of new technologies, solution complexity and increasing quality demand are among the toughest to be found in industry. In hyper competitive environment such as a software industry, possession of knowledge and using knowledge efficiently provides competitive advantage. Knowledge and skills are often personalized to individuals among employees and these capabilities are lost when knowledge walks out of the door (Ramanujan et al. 2004).

One of the key concepts of knowledge sharing in people business is the concept of a learning organization. This is the case company's company level strategic focus, which occurs as a mindset or is a culture inside the company. Garvin (1994) defines a learning organization as "An organization skilled at creating, acquiring, and transferring knowledge and at modifying its behavior to reflect new knowledge and insights". Garvin concludes five main activities for a learning organization: systematic problem solving practices, experimentation with new approaches, learning from their own experience and past history, learning from experiences and best practices of others and transferring knowledge quickly throughout the organization. This means that experiences should be gathered, made available for the whole organization and utilized actively.



Companies invest differently in education and learning. The most typical investments in software industry are training courses and R&D or demonstration projects. Demonstration projects are a very good example of learning investments inside the company. Such a project aims to solve some problem, get information of a technology possibly suitable for the company to utilize or gain some competence lacking from the company. Many times it can be useful to create a demonstration as a part of a tender to convince the customer and to visualize the solution. It probably is essential that demonstration projects or R&D projects are documented and experiences available for anyone in the company to use as a learning material. Objects learnt can be transferred in many different ways; written, oral, visual reports, site visits, tours, personnel rotation, education, training and standardization (Garvin 1994:8).

Continuous fast learning is one of the top priority issues to maintain high-level competencies. Competencies of the personnel of the company are building blocks of competent and competitive Company. Traditional individual or group learning is considered to too slow. Learning on organizational level and capitalizing organizations knowledge assets become very valuable for software companies. (Althoff et al. 2000:1). In a project-oriented industry such as the software industry, the largest obstacle to create learning organization can be constant hurry; management should be able to free some time to study new, possibly beneficial technologies and to learn new valuable competencies. Another obstacle can be information culture in the company. The best results are achieved if the culture is supportive and open. In short, to be able to utilize knowledge, knowledge should be managed.

To be able to learn and reuse accumulated knowledge, project results, success reports and review results should be represented in a uniform way. This means that experiences from projects should be gathered into a system, which presents the project in the same way. Having projects and experiences formalized like this enables the comparison of the projects. This kind of system should have good search features and easy to use editing tools. The presentation should have structure, which starts from a holistic view and enables drilling into the more detailed parts. Wiki can be considered a suitable tool for acting as an 'experience base'. Projects should analyze and document the lessons learned and gather quantitative, relevant figures to describe and pass on to others information about how successful the project was.

Figure 8 illustrates one example of defined experience environment architecture.



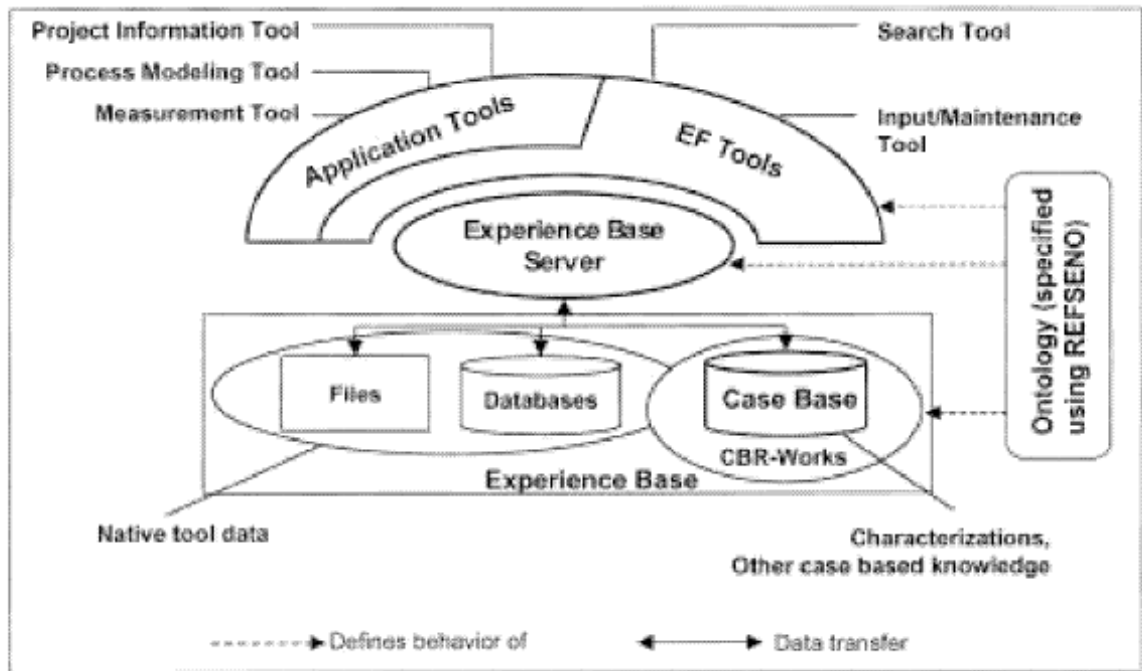


Figure 8. Architecture of the software engineering experience environment (Althoff et al. 2000:12).

In Figure 8, the core of the experience environment is the database together with documents. The environment has then Case Base, which enables knowledge sharing between projects and learning from previous projects. The application server then uses the data to provide applications and tools for the company using the environment.

The central objects of Knowledge Management include documentation of tacit knowledge of experts, creation of electronic repositories to store knowledge and use of tools for electronic collaboration and data searching. Achievements in different companies vary by size. Larger companies have more complex and advanced systems and smaller companies have more moderate solutions. In the larger end of the scale, Knowledge Management systems are feature rich, tailored for the targeted company and their global business. Smaller companies typically tend to use Wikis. Wikis have inbuilt, basic features which support the objective; search functionality, easy to use editing, project or competence focused spaces, user or user group restrictions, blogs and so on. What Wikis do not have is structured data models or structures to store knowledge related or competence information. Structured models force knowledge to be more comparable, searchable and indexed format. On the other hand, Wiki's problem and at the same time strength is openness and relatively free possibility to document and use the information. When there is the possibility to input experiences and

information freely, a person using the system does not feel chained to the restrictions of the system, which might happen if data is strictly structured. Especially with developers, as artists and experts in the system creation, too formulated demand of input might be an obstacle for the documentation goal. As a principle, people are motivated to document things only if they see clear benefit out of the work, for their own purposes or for the common goal they are committed to. In rapidly advancing, challenge-facing projects, documentation for 'next generations' might be hard to achieve. Project performance goals might go over the responsibility to document one's knowledge. This is especially difficult to manage. When an organization has software development challenges, there is a tendency to include a more disciplined, detailed and audited process for the organization. This causes the situation to get additional deterministic controls on a dynamic environment. It generally makes a bad situation worse (Poppendieck et al. 2003).

### 3.7 Conceptual Framework of Process Design for Technical Solutions

The conceptual framework collects the most relevant best practices found from the articles and literature. It is a basis for actual model planning utilizing the gathered data. In this Thesis, the conceptual framework presents best practices as 'influencers' to the CMMI Technical Solution concept, which is used as a basis for the process development. The CMMI Technical Solution process consists of two Specific Goals before the actual implementation: Select Product Component Solutions and Develop the Design (Software Engineering Institute 2010:375). These elements are in the core of the conceptual framework.

First, there is an overall approach to be selected for the process development; more agile approach or more controlled and detailed approach. An agile approach tries to get benefits from speed, flexibility and focus. A controlled or detailed approach, on the other hand, tries to get benefits from top quality, audits, detailed documentation and careful planning. In this case, the agile approach was selected as a starting point for the conceptual framework. A stimulant for this was Poppendieck's observation that when organization has challenges, a more disciplined, detailed and audited process could cause troubles instead of improvement (Poppendieck et al. 2003). Another stimulant was Lean methodology's first principle to remove all waste from the processes, which can be interpreted as removing unnecessary documentation and heavy auditing from the process.



Influencers are selected by suitability and applicability to the Technical Solution process. There is a large amount of practices overlapping in these best practices, which can emphasize the importance of the individual practice. However, it can be recognized that there can be core practices in frameworks, which are not suitable to more agile and appropriate process. One of these is the Six Sigma framework, which emphasizes high quality through strict, audited and measured processes. The Six Sigma approach seems to be more suitable for manufacturing than for the Technical Solution process.

PSP (Personal Software Process) seems to have a useful concept of design reviews. In software design, reviews by colleagues can be fruitful mechanism to have a second opinion about the design. PSP also defines Design templates, a sort of reference designs to be used as a basis for new designs. These templates are collected from projects done before, which are linked also to the objective of learning organization and knowledge sharing.

The main principle in Lean methodology focuses on 'waste' in processes. Removing unnecessary procedures from the process can be thought to be cleaning waste from the process. Lean also includes the principle of making decisions as late as possible. In software development, making detailed decisions as late as possible is essential since definitions are typically not accurate enough to cover all the details of the environment or procedures. The Lean principle 'See the whole' is also strongly related to software system design and especially to software architecture; today's systems are very complex entities having dependencies to numerous other systems, services, data sources and applications. It is very important to have knowledge and awareness of surrounding influencers. Lean emphasizes also involving the team. In the Technical Solution process, this can be applied in a way that the implementing team is participating in the design as much as possible, which is related to knowledge sharing within the team as early on as possible. Individual experts are the best resources to discuss with about the solutions in the design.

Agile methodologies have a principle of evolutive solution; i.e. the solution evolves during the project, which can be interpreted, that it might be not useful to do design on a very detailed level at the beginning of the project.



The ADD approach recommends communicating more by pictures and diagrams instead of huge amounts of written text. This might be a good approach to share knowledge and discuss the design with colleagues to have second opinions.

Six Sigma, although being quite heavy and probably not so suitable for a design process, contains a principle of open communication. This can be a valuable principle related to knowledge sharing, agility and learning organization.

Knowledge sharing and learning organization are overall goals related to any processes. This approach is important to take into the conceptual framework.

Figure 9 presents the conceptual framework of this study.

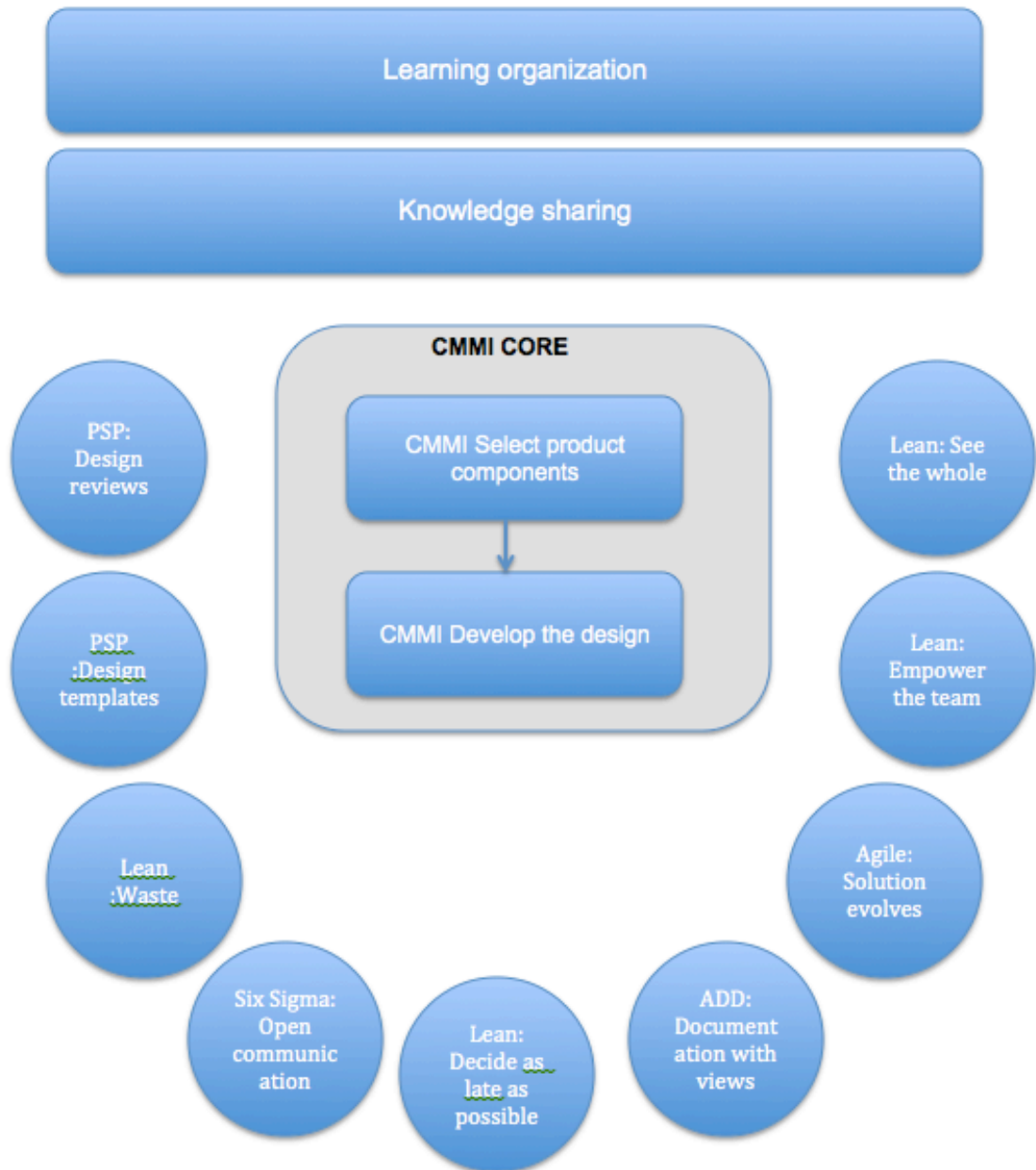


Figure 9: Conceptual framework for this study.

In Figure 9, Core brings a basic structure of the CMMI Technical Solution process in to the conceptual framework. Other findings from different frameworks presented earlier in this study are considered to be *Best practice influencers*. Top-level goals are knowledge sharing and learning organization.

## 4 Analysis of the Data

This section describes the data collection of this Thesis. It describes the approach selected, the inquiry made in the case company and the results of the interviews made.

### 4.1 Approach

As section 2.2 of this study presented, approach to collecting data was decided to be two-fold: an overall inquiry available to the entire personnel to participate and targeted interviews to get more detailed opinions and visions from the selected professionals inside the company. The questions were planned based on best practices found from the literature. The main themes in the inquiry were overall vision and opinions on software planning, knowledge sharing, learning organization, re-usability, alternative solutions and reference architecture templates. The themes in the targeted interviews included optimal design process, proper detail level of design, structure of technical plan, agility and product oriented planning process. There were also prepared questions on agile procedures related to removing the waste, documentation by views instead of text, design reviews and open communication inside and outside the team. The questions related to re-usability and alternative solutions follow the principles of CMMI. The level of the design details and the structure of the plan is related to agile methods, Lean principle of removing waste and Lean principle of deciding as late as possible while documenting by views is related to ADD-principles. Open communication is one of the principles in Six Sigma.

### 4.2 The Perspective of All the Staff: Inquiry

When asked about the best practice for solution design, the respondents felt that there was a clear need to have a decided and *defined process* to follow and that this process should be *monitored by audits*. It was thought that the company has a quite *clearly selected set of technologies*, products and software *frameworks*, which are guiding the design and competencies needed in the projects. There was a special need to have more guidance and selected solution paths to more detailed problems. In this chapter, there are in-text quotations presenting actual opinions of respondents.

Guidance could be for example *suggestions for software libraries* to be used for solving a certain kind of a problem.



There could be knowledge available to solve some specific problems. Description could also point in which project this problem was solved and how (Inquiry response).

Still it seems that designers need to have some sort of *freedom to use* the best possible solution to a detailed problem. A component or library suggested by the process cannot be mandatory.

There should always be some level of freedom to choose the best component or library even if it is not used before in the company (Inquiry response).

These specific problems are considered to be anomalies in projects, which should be still controlled. Freedom to select the best possible solution is also considered to be an innovation-enabling way of work whereas too tight a solution environment is considered to be too restrictive. It is probably felt that best practices or the best process should be easy to adapt and understand.

Clear process should not automatically mean chaining the creativity in the design (Inquiry response).

The process should be clearly communicated and all information should be available in the company Wiki.

When respondents were asked how we should better share knowledge and how we should assist the development of the organization, there was a clear need to have projects, service portfolio and experiences to be *documented* in the company Wiki in a clear, structured way.

All knowledge having learning value should be stored into the company Wiki (Inquiry response).

*Technical audits* should be *openly available* to enable the organization to learn about mistakes or successes inside individual projects. There was also a need to *organize 'Tech Afternoons'*, informative events to communicate recent learning, findings and best practices. Re-usability was one key issue in the inquiry.

Possible re-usable components should be recognized as early as possible (Inquiry response).



The answers show that there was a need to develop own component libraries to support project work . Therefore, there should be pages in the company Wiki presenting readymade component libraries. One good idea was to include *recognition of re-usable components* in projects as early on as possible. Knowledge sharing over team boundaries and projects was seen to be a very important improvement. A business oriented vision was that projects should have focus on evaluation if there is a possibility to develop the product from the result of the project. This evaluation should also be done as early on as possible. This consideration should be active throughout a project, from design to delivery. *Proper documentation and audits* were considered to be very important in this goal. It was also seen that thorough familiarization with literature and articles covering some important topic should be done when such an opportunity occurs.

When respondents were asked how the Company could develop re-usability, it was thought that it is not trivial just to decide that the company will create re-usable components. There seems to be two approaches; either to design new re-usable components in projects or find re-usable component frames from projects done before. In the second option, components should be developed by refactoring the produced modules, created in the delivered projects. However, this is always an investment. There was seen one major problem; definitions tend to be changing during projects, which means that well planned, re-usable components can end up being very customer specific. There was a common opinion that hectic, fast advancing projects cannot include refactoring of these components to be generic in their revenue targets since refactoring is a cost and schedule issue. The Technical Solution process should support the creation of re-usable components by enabling investment injection in cases where creating re-usable modules is a possibility.

On the whole, knowledge sharing and learning from previous deliveries was seen to be very important, motivating and effective.

Technology wise, most generic, standardized technologies were seen to be the most reasonable choice in terms of motivation, continuation and flexibility of resourcing.

When asked which technologies developers think will be successful in the future and which technologies are the most motivating, the feeling seems to be that the company's visions and focus are currently about right. So-called 'legacy' technologies were recognized and in line with what the company has already decided.



*Comparison between solutions* was seen to be most fruitful if there is an auditing developer to discuss the solution.

Couple of developers should get together, evaluate candidate solutions and discuss pros and cons (Inquiry response).

In comparison, structured, easy to find reference projects were considered to be very important. In the preliminary phase of the design, it was seen that a comparison should be done instead of just quickly deciding on the solution. It should be noted that the variety of technologies used should not be too wide. Still, if there is an even better solution recognized outside the best practice solutions, choosing the best one should be allowed.

Continuous learning was emphasized. The lead developers should have a common vision of the recommended solutions. Yet again, lack of knowledge sharing was felt to be the problem at the time.

Finally, the responsibility to search and develop re-usable components and document experiences was considered to be a key performance factor in the future.

#### 4.3 Lead Developer Perspective: Interviews

The interviews were targeted at the Lead Developers of the company, responsible for the application architecture. A larger project might have two or more Lead Developers; each specialized to some layer of the architecture. Lead Developers are always also producing code and components to applications; they have a long experience and visionary attitude to the solutions. When interviewees were asked what features an optimal design process would have, which is emphasized; Agile or ADD and what the needed level of design is, most of the interviewees thought that the company should prefer Agile over a very detailed design at the beginning of the project.

Probably there has been little bit too much designing at the beginning of the projects (Interview A).

This result supports interpretations made in the conceptual framework. It was seen that the design almost always changes during the project; only significant, guiding products, frameworks and components should be designed at the beginning. This result is in line with the Lean principle 'Decide as late as possible'. On the other hand, if the target is very clear and the definitions are carefully done, the system can be designed in as detailed level as possible. This variation in answers indicates that projects are differing from each other and there is no one clear level of design to be defined. A common



need is anyhow that at least guiding products, frameworks, development principles and components must be decided before the implementation starts. The developing team should be able to decide these together and no one should deviate from these decisions. Decisions made together were seen to be motivating but still the Lead Developer should have decision power and the responsibility.

As a documentation tool, the *company Wiki* was clearly seen to be the best tool instead of creating formal documents. Wiki is seen to be a dynamic, incrementally updating information tool available during the whole project. Documentation should be gathered to Wiki and it should be targeted to the development team as the most important stakeholder to use design documentation.

Flexible documentation tool enables sharing of documentation responsibility. Anyone can make additions easily. (Interview B).

It was thought that a *customer version* of a design should be separated as an individual, more targeted document. Every Sprint or section review should have a *checkpoint* to see that the documentation is in a proper level according to the needs of the team. The respondents thought that there should not necessarily be any strict template for design documentation but preferably a *checklist to remind* the Lead Developer about the necessary aspects. Documenting a design dominantly with diagrams and pictures was seen to be more practical and lightweight than writing boring, detailed text about the plan. This is in line with the finding from the literature; the ADD principle of using 'views'. It was also thought that the development team eventually cannot document every detail and it is not appropriate; the code itself and the tests related to the code helps to document the functional details in a more practical way than a written description

Good coding habit includes commenting the code. It is then describing itself without the need to describe it separately (Interviewee C).

*Planning of re-usable components* was seen to be reasonable if the plan contains suggestions instead of mandatory decisions at the start of the project. It was thought that during the project there appears to be functionality, which reveals to be re-usable, at least within the project scope. Therefore, it might be difficult and useless to define re-usable components at the design phase of the project. The components should be created in a way where a part of the implementation is refactored afterwards to produce the final re-usable component. The components planned to be used in the project,



should be listed in the detailed technical plan but the list should be more focused on containing suggestions rather than stone carved decisions.

#### 4.4 Summary

To summarize, the needs in the Technical Solutions process gathered from all the staff and Lead Developers communicate this: A defined design process is needed. The company should create a knowledge sharing environment or use it more widely (The Company Wiki). Learning should be ensured by getting experiences from past solutions. A comparison of the solutions is valuable when it comes to finding the most suitable solution. The creation of re-usable components is essential to reach a well performing organization. The company should have a lightweight, agile way to design. Mandatory documentation requirements should be removed or at least reduced. Documentation should be done in a way that it is motivating. Moreover, the development team should be considered to be the most important stakeholder to use the Technical Solution documentation. A software development process is an incrementally advancing learning process so therefore the design should also go incrementally to a more detailed direction during the project.



## 5 Building the Technical Solutions Process

This section builds the first version of the future Technical Solution process of the case company. It then summarizes the results and conclusions combining best practices found in the literature and the data collected. Next, it considers the future process from the point of view of different stakeholders. Finally, it presents first version of the future process.

### 5.1 Analysis of the Stakeholder Perspectives

The Technical Solutions process has a natural structure of four phases, as defined in CMMI for Development: Preliminary design, Architectural design, Detailed design and finally the Implementation phase. These phases are not clearly isolated; they can be mixed differently.

Solution design is clearly an iterative process where typically the first phase, Preliminary design, is made at the bidding phase or at the analysis phase of the product development project. The details and depth of the preliminary design depend on the entity to be offered and the requirements in the request of the proposal received from the prospect or customer. Since many times at the bidding phase there are no detailed definitions available, the preliminary solution has to rely on more general guidelines such as a company decided technologies or some similar, previous project having the solution design made. Figure 10 shows the basic Technical Solution process integrated from CMMI for Development.

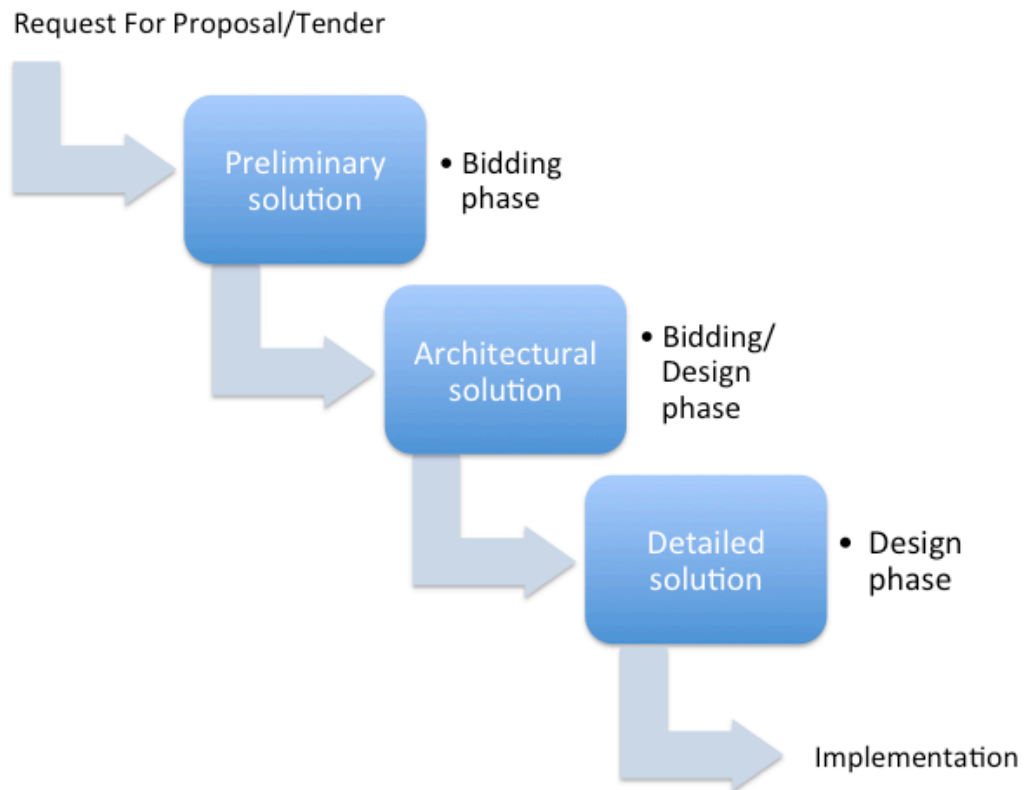
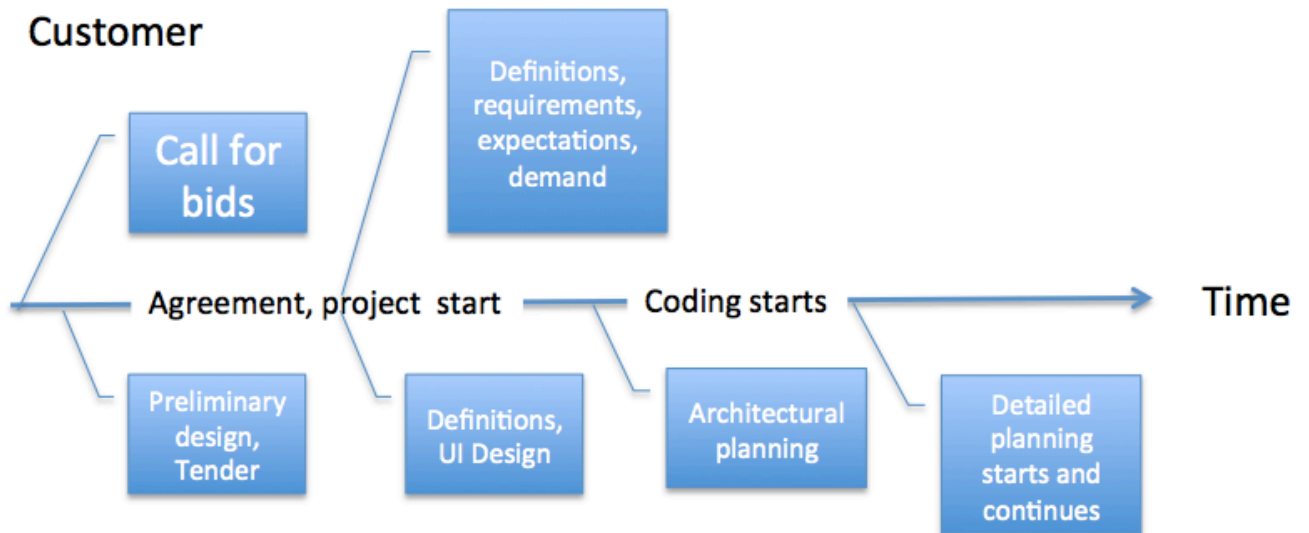


Figure 10. Basic Technical Solution process integrated from CMMI Technical Solution process area.

In Figure 10, the first phase in the Technical Solution process is Preliminary solution. Typically preliminary solution is made at the bidding phase as a part of a tender. The next phase is the Architectural solution, which is created if the bid has been successful. The third phase is the Detailed solution phase which is practically the detailed design phase.

Figure 11 presents the typical timeline in a project. The three phases presented in Figure 10 are hooked to this figure.



## Provider

Figure 11. Timeline in simplified, typical project.

In Figure 11 the timeline begins with the customer's call for bids. The bids are made and a preliminary design is presented as a part of the bid. The next spot on a timeline is probable agreement with the customer and the project start. The project starts with definitions (specifications), requirement documentation, expectations management and clarifying the demand. After these steps, the Technology Solution process parts follow.

There are different stakeholders that are going to be either related to or using the new Technical Solution process. The following sections describe these stakeholders, their motives and purpose of the process for them.

### 5.1.1 Management Perspective

In some cases, too detailed planning has been a waste of time and efforts and in some cases planning has been disregarded. Especially in preliminary planning, there is a possibility that too detailed planning is a waste of time; if the bid is rejected, efforts are wasted and in some cases, the prospect gets free of charge knowledge and ideas from the company. The planning should be on an optimal level with regard to both time and efforts. Too much unnecessary planning causes costs whereas too lightweight planning causes risks. Risks should be avoided because they cause unplanned costs. The purpose of the process should be that projects are planned in an appropriate level.

Knowledge gathered in the bidding cases should be shared and used in the future solutions.

### 5.1.2 Project Manager Perspective

Project manager (PM) has the main responsibility in a project. PM should take care of the fact that the planning is made after definitions phase and that the production or implementation should not start before the planning has been made and consulted (audited). Consultation should be documented so that the Lead developer has a clear list of how to improve the design so that generic, experience based risks can be avoided. The PM has to be sure that planning is made in appropriate level and that the documentation of the implementation will be made during the project. PM is also interested in team's motivation considering documentation. Documentation of plans and implementation should be easy and avoid unnecessary phases to ensure that the Lead developer, individual developers and experts are motivated enough to document the solution in appropriate level. PM ensures that when the project is done, documentation for the maintenance phase is made.

### 5.1.3 Lead Developer

The Lead developer (LD) is many times the same person who has planned the solution. In the planning phase, it would be very useful to have knowledge of the reference projects and what problems these projects had, encountered or solved. In the implementation phase, LD is responsible of the project going in the way it has been planned. The Lead developer is also making decisions considering the structure or solution of the end product. LD is a more experienced developer who supports other developers in their challenges during the implementation. LD makes sure that the plan is followed and that the company's selected technologies, frameworks and components are used. An individual project should not take new, unknown components to be implemented because that kind of individualistic behavior can cause unexpected technological risks. LD is also responsible for the creation of appropriate technical documentation during the implementation. The new process provides the LD with a straightforward way of working to achieve the goal of the project.



#### 5.1.4 Developer or Expert Perspective

A developer is coding the project in a team. The Lead developer is guiding the developer's work and dividing responsibilities of different parts of the project to the developers. The new process describes the planning process for the individual developer or expert. It would be very useful to easily find solutions to specific problems that projects are encountering. In many cases, previous projects have had similar problems and if these problems and solutions would be available somewhere easy to find, it would ease up problem solving situations.

#### 5.1.5 Customer Perspective

In most of the cases, the customer needs to have the preliminary planning made and described in a document to be used in the decision-making. In some projects the customer's IT department needs to have a more detailed technical solution described and approved before starting the development. In the cases where the customer has lined some products or technologies to be used to avoid a situation where individual applications are produced using different technologies, a more detailed technical solution has to be given and argued. The customer is also interested in the life-cycle situation of individual technologies. Technologies having no proven long life cycle are usually avoided. Moreover, technologies causing high yearly license fees can be rejected in some cases.

### 5.2 Proposal for a New Process

The proposed new process is based on the conceptual framework, presented in subsection 3.7. The conceptual framework was adapted to the case company's processes by comparing it to the current situation and practices. The overall principle was to develop the process to a more agile and appropriate direction by utilizing the visions gathered from the interviews. Agile clearly seemed to be the most reasonable direction considering the size of the company and the results from the interviews. The process was considered to be the best possible if it serves different stakeholders in different levels. It was decided that the main stakeholder would be the engineers, because they are the ones actually designing and developing the applications. Thus, the process is primarily company internal and for the people working for the company themselves.



Having involved people from all the stakeholder groups it can be expected that people will be motivated enough to follow the proposed process.

The following graph (Figure 12) is an illustration of the future process.

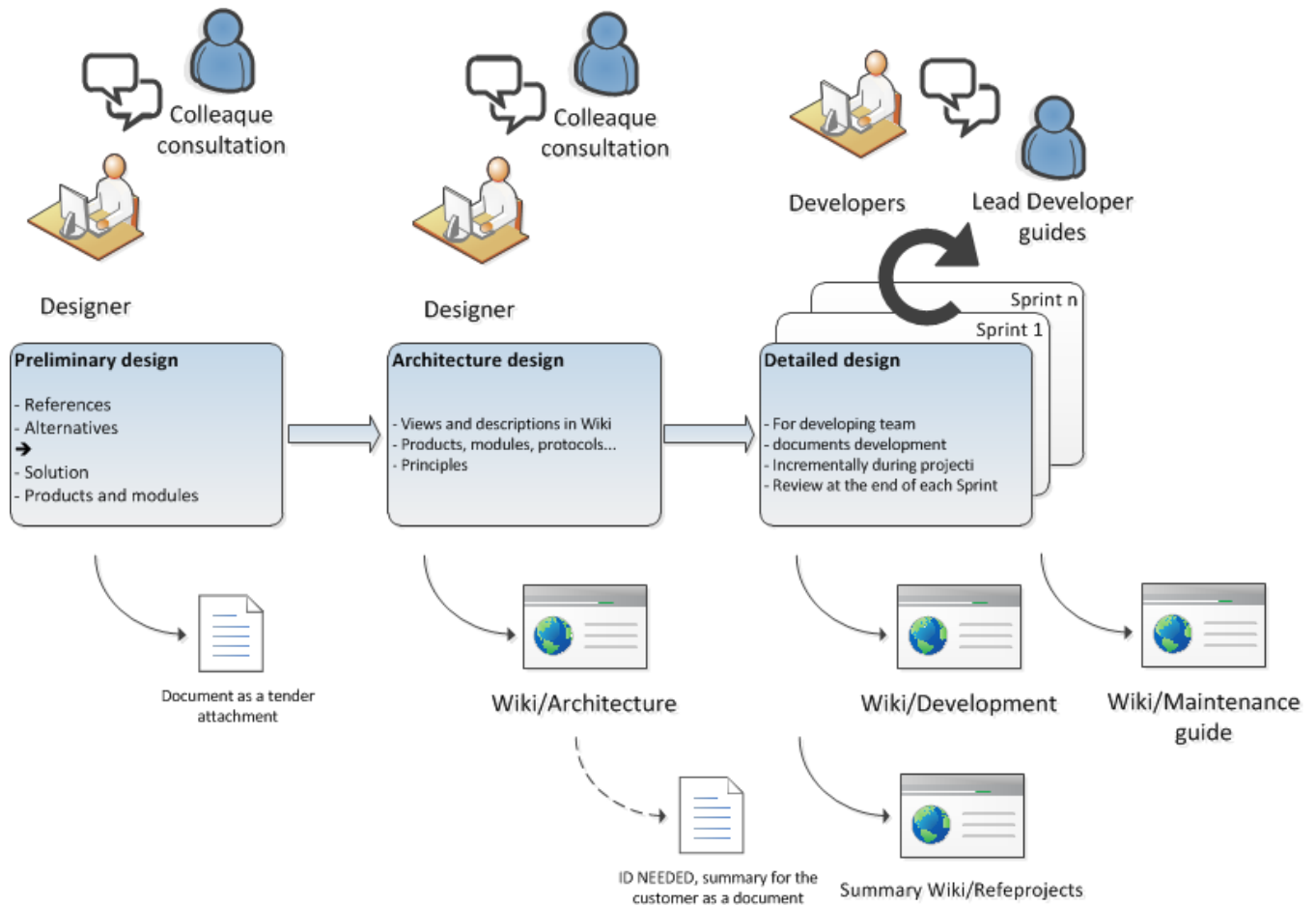


Figure 12. Proposed process.

In the proposed process there are three major phases as presented before in Figure 10. *Preliminary design* is made in the bidding phase and the output of that phase is an attachment to the tender describing the preliminary design. To have some out of the box considerations, there is a colleague consulting on the design and giving ideas and comments. The second phase is the *Architecture design*, which deepens the design made in the preliminary phase. The output from that phase is a Wiki page under the project space describing the Architecture. If needed, a separate summary is written for the customer. The third phase is the *Detailed design* which is an iterative process going on through the whole project. The output of this phase is a Wiki page Development

under project space. The process is designed to be agile with control enabled by colleague consultations. Each phase will be documented to the Wiki so that all the information is searchable and easy to find. This enables knowledge sharing and formulation of the learning organization. The focus in all documentation is on the developing team, the motivation comes from the fact that the documentation is created mainly for the team itself responsible for the documentation.

In the following the phases of the process suggested in Figure 12 will be explained in detail.

### 5.2.1 Preliminary Design

Phase 1 is the Preliminary Design. A typical situation where a preliminary design is needed is either the concepting phase of a new product, Request For Proposal (RFP) from prospect or tender creation (bidding) phase. As a basis for preliminary design there is always some knowledge about the need. Good example is customer's requirement definitions as a part of the RFP or call for bids. Chief Technology Officer, Technical Architect or Lead Developer participating tender creation makes the preliminary design. In larger bids a solution team is formed. It consists of experts having a certain, focused area of specialty.

There are several best practices presented in the conceptual framework influencing this phase. Knowledge sharing tools or methods can ensure that the person considering an approach to the solution can find similar cases and solutions. Knowledge thus found can also provide valuable information on which approaches are suitable for different kinds of cases. Some cases can require for example a commercial third party product while some other cases require the project to start from scratch. The design reviews can be utilized to ensure that alternatives are compared and the decided approach is not only one person's opinion. The design review can be used to get a consultative opinion on how to approach a solution. Using the Lean principle 'removing the waste' from the process can speed up the preliminary design phase, which is many times done in a tight schedule demanded by the RFP. The Lean principle 'to decide as late as possible' can be used to guide in making good preliminary decisions and leaving detailed decisions to the later phases. The ADD principle to use views in the documentation rather than lots of text can make the consultation of the solution much easier since the person giving the consultative opinion can very quickly understand the pro-



posed solution. The Agile principle 'solutions inevitably evolve in the implementation' leads in to the conclusion that the preliminary design should only define the basics of the solution and not to go in too deep into details. The Lean principle 'to empower the team' can be used so that for specific solution challenges, proper experts should be consulted. Still, the problem is that only in rare cases same resources used to make a preliminary design are available if the project gets started. What is very important is the Lean principle of 'seeing the whole'. In the bidding and preliminary design phase it is important to be aware of different factors and elements influencing the prospect decision making and create the design to fulfill different needs in the limitations of the influencing elements.

In the preliminary design phase there are several alternatives that can be thought of as an approach to the solution. A preliminary design can be considered based on the following aspects:

Aspect	Name	Description
PD-A1	Products	What products can be considered in the solution? (e.g. Microsoft stack vs. Open Source)
PD-A2	Architecture	What kind of general architecture would be the most suitable in this case? (e.g. Web application, Desktop application, Mobile application, Hybrid application). What is the conceptual data model?
PD-A3	Integrations	What kind of related systems there are? (e.g. SAP ERP, CRM, Billing)
PD-A4	Devices	What kind of devices would be most suitable? (e.g. Laptop, Smart phone, Tablet, rugged PDA)
PD-A5	Users	What roles are there and what is the amount of users in different roles.
PD-A6	Conditions	In what conditions the solution is used? (e.g. outdoors, car, office, temperature, rain, lighting, network availability)
PD-A7	Limitations	Is there something in the main requirements that limits the solution? (e.g. customer prefers some products)

Table 3. Aspects in Preliminary design phase.



It is important to create at least two different alternatives for the solution in the preliminary design phase. The main principle is: “One indicator of a good design process is that the design was chosen after comparing and evaluating it against alternative solutions.” (Software Engineering Institute 2010:375).

Alternative solutions are first described briefly, mainly focusing on a general solution and the pros and cons of it. At this point, a colleague is called to provide consultancy and discuss the most suitable approach. An approach is then selected from the alternatives and improved by issues raised in the consultative discussion. It would be reasonable to have another discussion after refining and adjusting the selected solution.

When comparing the alternatives, following a selection criteria can be used when thinking and discussing what the best approach would be:

<b>Criteria</b>	
PD-SC1	<p><b>General suitability to requirements</b></p> <p>How this solution reflects to most centric requirements? Is there some part that does not fill the requirements and why this solution is still relevant or the best? What customer’s challenges or problems this solution solves? How end users experience this solution? What kind of lifecycle there is to be seen for this solution?</p>
PD-SC2	<p><b>Background knowledge of customer’s priorities (From sales)</b></p> <p>What we know about customer’s priorities or valuation focus? Has customer told something about priorities e.g. products or approaches they like? Is the price the most relevant factor in this bid?</p>
PD-SC3	<p><b>Costs of the solution</b></p> <p>What kind of license, development, support, service fee or hardware costs there are in this solution? Is this solution the most expensive or the cheapest possible? Is this solution too heavy for the actual need?</p>
PD-SC4	<p><b>Demanded schedule compared to the solution</b></p> <p>Can this solution to be delivered in demanded schedule. What time consuming risks there are? Is this the solution, which is the most quickly delivered?</p>
PD-SC5	<p><b>Performance and reliability</b></p> <p>Is this solution suitable to the user amount or increasing user amount?</p>

	What performance risks there are? Is this too lightweight solution considering business criticality of this system? How reliability shows in this solution?
PD-SC6	<b>Risks</b> What general risks this solution has?
PD-SC7	<b>Suitability considering company technology course of conduct</b> Is there something different in the solution considering company technology course of conduct? New technology: Why it is important to differ from the company line?

Table 4. Selection criteria when comparing alternatives.

As an outcome, it is reasonable to write alternatives to the company Wiki so that the decision trail leading to a certain, optimal design can be followed. To achieve this objective, it could be reasonable to create a separate space for the Wiki including descriptions of tenders, alternative thoughts and the solution. People considering solutions to tenders could then find similar tenders made before and follow the decision trails especially for those solutions made in the winning bids. This information could also include interpretations of the lost cases to see if any alternative solutions could have supported winning the case.

In a case of a tender, the final outcome is the Technical Solution document to be attached to the tender. In a case of product development or an R&D project, the alternatives and selection is described in the company Wiki. When making a decision of the selected approach, it is fruitful to also consult the sales department to obtain their point of view.

### 5.2.2 Architecture Design

In Figure 12 on page 47 the process has Phase 2, which is the Architecture design. The preliminary design acts as a basis for the architectural design. A selection made in the preliminary design usually aligns the products, their relations and type of usage. The architectural plan stays as a general level plan still going a bit deeper in the details than the preliminary design. The Technical Architect or Lead Developer designated to the project makes the architectural plan. In some cases the preliminary design can fulfill the requirements of the architectural design, especially in small or mid-sized projects. This is acceptable only if the designated Technical Architect or Lead Developer



agrees fully to use the preliminary plan. The preliminary design can also be such a lightweight description of the solution that it does not have any relevance for the Architectural plan. The project team cannot be required to be fully committed to the plan made by someone who is not a member of the team.

A typical situation in tenders or RFPs is that the tender and preliminary solution has to be made without proper requirements or definition documentation. Therefore, the architectural plan made after the definitions is in most of the cases more accurate than the preliminary plan (see Figure 11 Timeline in simplified, typical project).

The Lean principle of ‘empowering the team’ can be utilized more deeply since the team implementing the design is decided at this point. Using actual experts responsible for the implementation is essential; involving them in the architectural design engage people better. Knowledge sharing is very important at this phase. Documenting the architectural plans in a way that other designers can easily find the design and use it as a reference will assist towards both optimal design and objective of the learning organization. This is also related to Six Sigma objective of open communication.

It is very important to be aware that the architectural design is made in such detailed or deep level, as the project needs. If the project is large, complex or has risks related to used products or technologies, the plan should be made with more details considered. If there are very accurate definitions made and available, there is no reason why the solution should not be planned in as much detail as possible or needed.

Table 5 lists aspects to be considered for inclusion in the architecture design.

Architectural plan can contain the following aspects:

Aspect	Description
AD-A1	Architecture and components (pictures and diagrams in key role, descriptions)
AD-A2	Interfaces between components. Communication protocols and standards
AD-A3	Applications and modules in different devices
AD-A4	Modules which are potential to be re-usable
AD-A5	Potential open source, commercial or the company produced components to be used. Components planned to be produced in the project.
AD-A6	Technical risk analysis related to technology issues.
AD-A7	Resource plan for performance and reliability (hardware aspect included)



AD-A8	Analysis for security issues, description how security is planned
AD-A9	If there are components or products, which are not in line with company course of conduct, reasoning for those.

Table 5. Aspects in architectural plan.

It must be emphasized that the architectural design does not have to have all aspects documented; relevancy is also in this case a good guideline. It might be useful to form a conceptual data model or deepen it if it has been defined in the preliminary design.

As an outcome, an Architectural design is entered in the company Wiki, in the project space.

### 5.2.3 Detailed Design

Phase 3 of figure 12 shows the Detailed design phase. This phase utilizes the same best practices as presented in the preliminary design and architectural design phases. The *Lean principle* of ‘empowering the team’ is important because individual team experts are responsible for the detailed design of their field. They are also going to be responsible for documenting the implementation during that phase. Knowledge sharing is important also at this phase. The team members must share their findings and detailed knowledge within the team. Documenting detailed plans and especially problems, solutions, learning and findings at the implementation phase will assist towards reaching the objective of the learning organization. Developers struggling with the same problems in other projects can find help and solutions in this way. This is also related to the Six Sigma objective of open communication.

The Detailed design is made after the definitions. The Lead Developer and developing team make the detailed design together. At first, in the first design meeting the Architecture design is checked and discussed in the development team. Absences are listed and the depth of the detailed design is decided. Sometimes in definitions, there are already some screens or mock-ups designed for user interface and experience. The definitions are discussed to define the approach for the detailed design. The detailed design is extended to the needed depth, at least a data model or database structure have to be designed. Figure 13 illustrates the Detailed design advancing from the first design meeting to the iterative development phase.



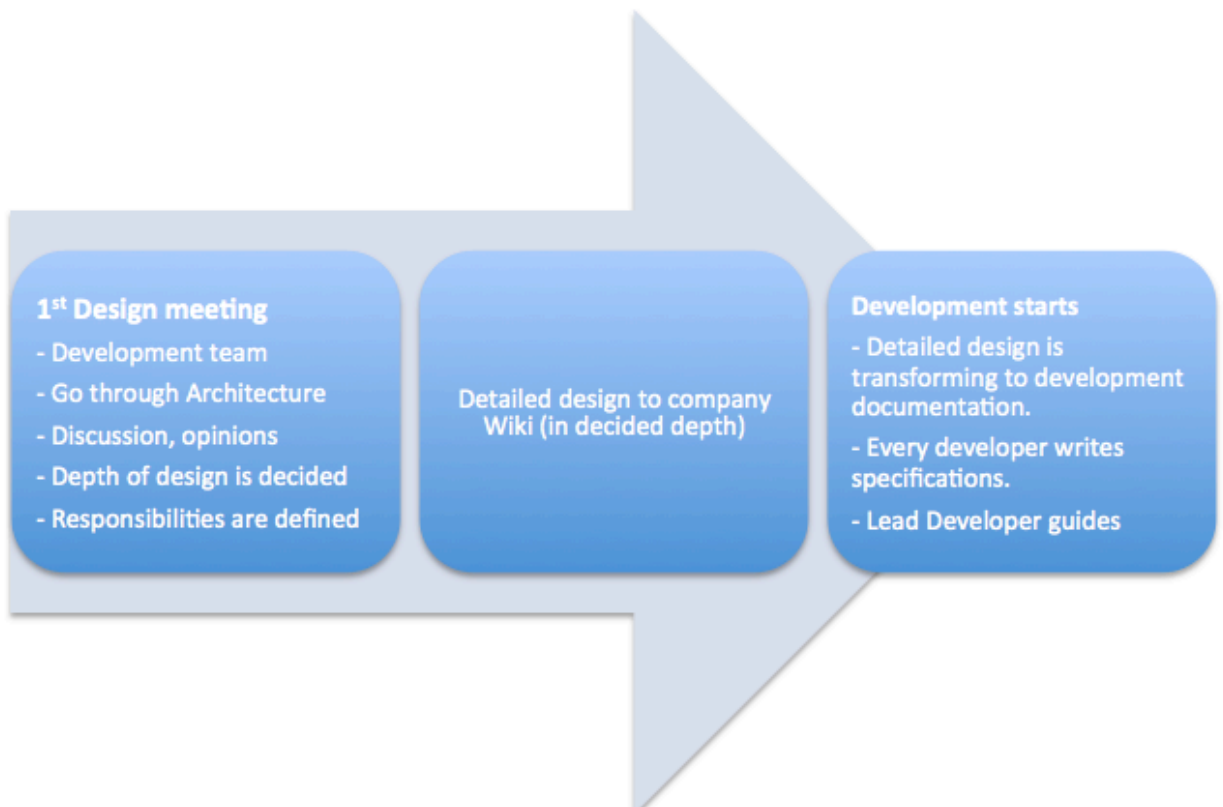


Figure 13. Detailed design advancing.

In Figure 13 the first design meeting is with the development team. All team members can discuss the design although the Lead developer is responsible for that. The responsibilities are defined inside the team. The next phase is to document the Detailed design into the company Wiki. At the next phase, the actual development starts and documentation continues all the way to the end of the project. The design is transforming to development documentation. Every developer writes documentation on the solutions or approaches they have. The Lead developer guides and checks that the team is documenting at an appropriate level.

The Detailed design is divided into several typical architectural layers. Web applications are applications provided over the Internet and used by Internet browsers. These applications have typically both a server and client end solution., Desktop applications are applications installed and run on individual workstations or laptops. These applications have typically just client solution but they can use services over the Internet. Mobile applications are applications used with mobile devices such as smart phones and tablets. These applications typically have both a client solution and server solution (services). Services are providing data to the applications over the Internet. The Data

model describes the structure of the data the solution is using. Integration services are services designed to join two or more systems together providing data for each other. It is reasonable to take into consideration each of these layers when designing the architecture.

In a detailed design some key measures are also designed. They relate to security, automated testing procedures and supporting product usage. Supporting products include e.g. Anti-virus and Monitoring products.

The Detailed design is documented in the company Wiki, under a specific project space where the Architecture is also documented. The Detailed design is an iterative process. Its depth and details of the design increase alongside with the advances of the development. At the beginning of the development, the following aspects are considered for planning:

Aspect	Description
DD-A1	Web applications (e.g. frameworks, components, design patterns)
DD-A2	Desktop applications (e.g. frameworks, components, design patterns)
DD-A3	Mobile Applications (e.g. frameworks, components, design patterns)
DD-A4	Services (e.g. web services, REST-services) and protocols
DD-A5	Data model and Database (Object model)
DD-A6	Integration services, Integrations
DD-A7	Security plan
DD-A8	Automated testing (policy in the project)
DD-A9	Supporting products (e.g. Anti-virus, Mobile Device Management, Monitoring)
DD-A10	Distribution planning (High Availability when needed)

Table 6. Aspects in detailed planning.

After creating a deep enough detailed design the actual development or starts. At this point, the role of the Wiki changes from plan to development documentation. At this stage every developer is responsible for documenting the parts he or she thinks is reasonable to be documented. The motive for documenting can be defined based on usefulness. If the developer, someone in the development team or someone in a future development project could need the information, it should be documented into the Wiki.



When using the agile development model, where the project is divided into sprints, documentation is necessary.

At the end of each sprint there is a development team discussion, where the level of documentation is checked. Every developer tells what has been documented and why.

Each developer also objectively estimates what should have been documented but was not documented. The Lead Developer acts as a facilitator of this meeting asking if there is something useful the team should specify for other people. Another important question is that is the code itself documenting the development enough or should there be supporting explanations in the Wiki. This is the question developers should have in their mind during the development. If the component is such that the code itself documents behavior and meaning enough, there is no need to do unnecessary documenting. Detailed design transformation to development documentation is purely for the development team, people continuing development in the future and people supporting the solution in the solution's future lifecycle.

#### 5.2.4 Development

This Thesis does not focus on the development process. Still, the development phase is related to the proposed model because development is always iterative and incremental; parts of the software follow each other. During the development and between iterations, the documentation is evolving. Finally, the detailed design has evolved to detailed documentation on how the software was done, what challenges were solved and what best and worst practises were found.

#### 5.2.5 Knowledge Sharing and Enabling the Learning Organization

In the current state analysis it was found that knowledge was divided in to two main sections; skills and knowledge of individuals and several data sources inside the company. Knowledge of individuals was rarely shared except in verbal, non-formal discussions. Documented knowledge was scattered to different data sources such as documents in the company file server, documents in the company version control system and descriptions in the company Wiki. It was a clear target to combine knowledge to a single centralized system having search features even if the needed documents only exist as links in the system to enable drill-in knowledge digging. The company version control had all the source code commented having the most detailed knowledge of the solution details.



The Company Wiki seems to be the best solution to be as the centralized hub for knowledge sharing. Wiki can have links to the documents in version control. The documents are opened immediately from the links without the user having to have any competence in using the version control itself. Wiki has search features to enable fast information finding within huge data masses. Wiki can have clear structures enabling reasonable drill-in procedures. The author can add pictures and multimedia into the Wiki pages to enrich the understanding.

It seems to be that there should be commonly agreed structures inside the Wiki to ease up knowledge finding and learning. The company already has a procedure for each project to have their own Wiki spaces. Suggested structure from the Technical Solution process point of view is presented in this table:

Space	Page	Usage
<i>Project space (each project has its own space)</i>	Architecture	Planned architecture. First view is the overall architecture having related systems and dependencies. Different parts or applications are described in separate views. Text is used to describe architecture in more detailed level.
	Development	Development time documentation. This page should be divided to proper subpages to avoid very long page.
	Maintenance guide	Guide for the people maintenance the delivered product. Should have list of 'most well-known issues'.
<i>Reference projects</i>	Main page for overall description of projects accomplished in the company. Table having each project in a one row. Each row has a link to the	To have quick overview what projects the Company have done, using which technologies and to solve what business challenges.

	project's space.	
<i>Tenders (Preliminary solutions)</i>	Main page for overall description of tenders committed. Table having each tender is a one row. Each row has a link to the tenders page in this space.	To find solution paths ending up to some decisions through alternative solutions documented. Tenders are describing Preliminary designs made in tenders having link to the Solution document sent within the tender document package.
	Each tender has its own page describing alternative approaches discussed.	To follow decision paths.

Table 7. Suggested Wiki structure.

In the inquiry and interviews made several challenges came up in both the preliminary design phase (tenders) and architectural planning phase. These challenges can be solved using the presented Wiki focused approach.

This kind of centralized documentation enables the following challenges to be solved:

<b>Challenge or question</b>	<b>Solution</b>
How to find similar projects we have done in the tender phase?	Similar projects can be found from the Company Wiki, Reference projects space. One can easily find similar projects and drill-down to the projects details through links.
How to find similar tenders we have made in the tender phase?	Similar tenders can be found from the Company Wiki, Tenders space. One can easily find similar tenders and drill-down to the tender details and alternative solutions thought through links.
How to find decision paths leading to some particular solution?	Tenders have description of the decision path and influencers on that.
What problems or challenges my approach to this solution might have?	By discussing with consultative colleague, these kinds of issues can be raised.
What best practices I should use in this	Similar projects have a list of best practis-



solution?	es found. These should be considered when planning the new project.
How should I share my findings in this particular solution others to utilize?	Structured Wiki-information, efficient usage of Wiki tags, Wiki search features.
Who are the people involved to similar cases I should consult?	Similar projects have team described in the presentation of the project (project page)

Table 8. Challenges to be solved by centralized Wiki documentation.

## 6 Implementation of the new process

This section describes experiences from the implementation of the new, proposed process.

The proposed process was individually presented to the piloting group including two project managers and two Lead developers. These projects were in the beginning of the architectural planning phase. The projects followed the suggested process and the Lead Developers gave their opinions about the proposed process.

The Lead Developers took the proposed process in general well. They thought that the proposed process is much better and well thought out compared to the old process. Still, there was some uncertainty over what to document and what should be the level of details in planning. This is the most significant difference to the old process. In the old process there was a document template, which was filled in for all projects even if it was a waste of time. In the new process, the power of the decisions considering planning is given to the Lead Developer; he or she should think what to document, how detailed and for what? This needs a change in thinking and is clearly an issue, which should be emphasized in trainings and presentations if the new process is adopted. Without the Lead Developers feeling both responsibility and interest, the process will not be successful.

It would also be reasonable to discuss with every Lead Developer when they start their first project with the new model. Especially, when the architectural planning is made, or should have been made, it could be reasonable to check out that the new process is well understood by both the Lead Developer in the project and the advice-giving colleague.



## 7 Discussion and Conclusions

This section summarizes the study and discusses the practical implications of the study.

### 7.1 Summary

This Thesis focuses on improving existing, undefined procedures of technical planning used in the case company. It presents a whole new process for technology solutions and software design. This improvement is needed to ensure that projects made in the company will use the same kind of principles, procedures and documentation while planning solutions for bids and production. The improvement has aspects of quality, comparability, flexibility and visibility to ensure that the organization is learning from past solutions and decisions, which makes it possible to advance in creating a learning organization.

The case company in this study is a software company developing both products and turnkey projects for its customers. Today, more and more of the deliveries are offered using the SaaS model, which means that the customer is using the product as a service rather than purchased licenses. The current procedure in technology solutions has been relying on document templates describing the needed aspects in planning and design. This has caused a situation where all the documentation is in individual documents, which is not the optimal situation in terms of knowledge sharing and learning. The case company has decided to put continuous efforts into process development and knowledge sharing. These decisions have been the guiding results of this Thesis.

The research approach in this Thesis is action research. In process development this kind of approach seems to be the most suitable. The improvements verified in the pilot projects are important to ensure the suitability of the created model. Proper research to find best practises, well planned data collection using inquiries, interviews and workshops and model building combining both best practises and the data is essential to find the most practical process.

The study starts with a current state analysis of the technology solutions practises in the case company. The company uses the CMMI framework as a basis for all process improvement projects. The results of the current state analysis were used to find suitable frameworks, models and practises to be utilized in the defined process creation.



The overall target was to have a well enough designed process to reach CMMI level 3 in the upcoming assessment. Using the current state analysis and findings from theoretical sources as a basis, data collection was started. The data collection was carried out inside the company through an inquiry targeted to the whole personnel, targeted interviews and discussion workshop. Combining best practises found from the theoretical research and visions gathered from the data, the first version of the future model was created. This prototype was then verified in two piloting projects. Feedback from these piloting projects was collected and used to improve the prototype of the model, which led to the final version of the model.

The new model will be more agile, suitable and defined than the earlier procedure. The new model will motivate individuals to document useful and appropriate details, enable knowledge sharing and lower risks in the upcoming projects.

## 7.2 Practical Implications

The new model was validated in two projects. The validation, or piloting, was important in order to have feedback on the practises planned. The feedback is used to improve the model.

To be used company-wide, the new model should be communicated in separate phases. First, the model should be briefly presented to the whole personnel in a suitable info event. It could be wise to first inform the personnel by entering a blog writing into the Wiki. The next step is to present the new model more deeply for management, Lead developers, Technical architects and Project managers, who are responsible for the model implementation. These actions should ensure that the new model would work efficiently and effectively in future projects.

## 7.3 Evaluation

### 7.3.1 Outcome Compared to Objective

The objective was to develop a totally new and suitable Technical Solutions process for the case company. The starting point was the situation without any defined process, just some procedures supported by document templates and customer specific ways of work. The main aim in process development was to find industry best practises by searching widely different frameworks and models and by truly listening to people who



are tied to this process almost daily; the people who are creating solutions for the bids and for the projects. It was recognized that planning is the very place to influence the success of the project.

Since this study was constructed in a way that there was first thorough best practice research, inquiry and interviews inside the company and the creation of the model after these, it can be interpreted that the outcome should be valid. A very important aspect is that this study focuses on creating the first version of the model, although it had feedback from validating or piloting projects. Process improvement must be thought of as an on-going, iterative cycle changing the model all the time to the more suitable direction.

The feedback has been encouraging. The people involved seem to have an opinion that this process area is very important and the results so far are encouraging making solution creation and design more motivating and meaningful. There have been comments that this is the first time in the case company that solution making has the focus it deserves.

### 7.3.2 Reliability and Validity in this Study

To secure its reliability, this study carried out a companywide inquiry and targeted interviews for lead developers as a data collection method. For this kind of focused process, this can be thought to be a broad range of data. The findings were discussed in the company Tech Steering Group. One can say that almost all relevant technology oriented persons in the company were participating in this study somehow. The researcher has worked for the company for several years and has thorough knowledge about the processes, objectives and the current situation in the company. Such knowledge is a strength because it ensures that the improvements found are relevant and suitable for the company. On the other hand, when the researcher is an outsider not working for the company, it enables a more objective approach for the study. In this sense, experience can also be a weakness.

The Lead Developers in the validating projects thought that the new process is much more suitable and well planned than the old process. The new process is planned to suit all projects in the company. Therefore, the results of this study are transferrable to all projects in the company.



The inquiry had seven respondents and five persons were interviewed for this study. A larger amount of respondents and interviewees could have provided more reliable data. The respondents and interviewees were all employees of the company. It could have been valuable to gather data also from outside the company, from other companies in the software industry.

This study enables a new way of software design in the case company. It has been rewarding to hear that lead developers are satisfied and motivated about the new process. Topics such as relevant, meaningful work and learning are considered to reflect the results of this study. Motivation to act by the process is one of the key success factors of change management.

The study relates to global themes in the software industry; quality, agility and effective knowledge sharing. These themes are considered to be key success factors in the software industry currently and in the future.

## References

### Articles, Journals:

Althoff, K-D, Bomarius, F., Tautz, C.(2000). Knowledge Management for Building Learning Software Organization. Information systems Frontiers Oct/Nov 2000 349-367.

Braun, E.(2005). Lean/Agile Methods for Web Site Development. Online (magazine) Sep/Oct 2005, page 58.

Cyrillo, M.(2011). Lean UX: Rethink Development. Dr.Dobb's M-DEV Nov. 14, 2011, pages 40-44.

Fichman, R.G.,Kemerer, C.F.(1997). Assimilation of Software Process Innovations: An Organizational Learning Perspective. Management Science Journal Oct 1997 1346-1363.

Garvin, D. (1994). Building a Learning Organization. Business Credit. Vol. 96 (1), 19-28.

Golafshani N.(2003). Understanding Reliability and Validity in Qualitative Research. The Qualitative Report Volume 8 Number 4 Dec 2004 pages 597-607.

Healy, M.,Perry, C.(2000). Comprehensive criteria to judge validity and reliability of qualitative research within the realism paradigm. Qualitative Market Research: An International Journal Volume 3, Numer 3, 2000, pages 118-126.

Hinkle, M.(2007). Software Quality, Metrics, Process Improvement and CMMI. IT Pro Magazine May/June 2007 47-51.

Hosley, S., Lau, A., Levy, F., Tan, D.(1994). The Quest for the Competitive Learning Organization. Management Decision vol 32, No 6, 1994, pages 5-15.

Jakobsen, C.R., Johnson, K.A.(2008). Mature Agile with a twist of CMMI. Publication of Agile 2008 Conference.

Karlström, D., Runeson, P.(2005). Combining Agile Methods with Stage-Gate Project Management. IEEE Software, May/June 2005, pages 43-49.



Middleton, P.(2002). Lean Software Development: Two Case Studies. Software Quality Journal, 9, 241–252, 2002.

Morse, J.,Barrett, M., Mayan, M., Olson, K.,Spiers J.(2002). Verification Strategies for Establishing Reliability and Validity in Qualitative Research. International Journal of Qualitative Methods 2002.

Nord, R.L., Tomayko, J.E.(2006). Software Architecture Centric Methods and Agile Development. IEEE Software (Journal), Mar 2006, pg. 47-54.

Ramanujan, S. and Kesh, S.(2004). Comparison of Knowledge Management and CMM/CMMI Implementation. Journal of American Academy of Business, Cambridge Mar 2004 271-277.

Rao, K.P.(2010). How can CMMI Help Six Sigma. Software Quality Professional Mar 2010 28-38.

Rubinstein, D.(2007). Practice Lean Development? You Bet You Kan(ban). Software Development Times Oct 15 2007 page 8.

Rupa, M.(2005). Six Sigma for Software. Software Quality Professional, Dec 2005 12-26.

Rupa, M. (2011). Software Six Sigma and Cultural Change: The Key Ingridients. Software Quality Professional, Mar 2011, pg. 38-48.

Santos P.(2007). SCRUM Meets CMMI. Dr. Dobb's Journal Sep 2007 28-34.

Schneider, K. and Von Hunnius, J-P.(2002). Experience in Implementing a Learning Software Organization. IEEE Software (Journal) May/June 2002 46-50.

Swart, J., Kinnie, N.(2003). Sharing knowledge in knowledge-intensive firms. Human Resource Management Journal, vol 13 No 2, 2003, pages 60-75.

West, D.(2009). Agile Processes Go Lean. Information Week, Apr 2009. Pg. 32-36.



**Books:**

Ahern, D.M., Clouse, A., Turner, R.(2008). CMMI Distilled. Addison-Wesley.

Eisenberg, E.M., Goodall, H.L.(1993). Organizational Communication. Location: St.Martin's Press.

Ferrance, E.(2000). Action Research. Location: Brown University.

Garcia, S., Turner, R.(2007). CMMI Survival Guide. Addison-Wesley.

Holcomb, M.(2009). Running an Agile Software Development Project. John Wiley & Sons. 220 pages.

Humphrey, W.(2005). PSP. A Self Improvement Process for Software Engineers. Addison-Wesley.

Poppendieck, M., Poppendieck, T.(2003). Lean Software Development. An Agile Toolkit. Addison-Wesley.

Software Engineering Institute(2010). CMMI for Development, version 1.3. Publication of Carnegie Mellon University. 482 pages.

Tomal, D.R.(2010). Action research for Educators. Rowman & Littlefield Education. 197 pages.



## Appendix 1: Questions in the Inquiry

Inquiry was kept using Finnish language, these are translations of the original questions.

1. Do you make technology related decisions in your work e.g. for tenders or implementations?
2. How important you think process improvement in the area of Technology Solutions is?
3. What kind of way of work is the best when creating design?
4. Do you know Company technological course of conduct?
5. How does the re-usability works is the Company?
6. Tell your idea how we can better share knowledge and how we enable the learning organization.
7. What technologies you think are future technologies.
8. With what technologies you would like to work with?
9. How we can better utilize earlier projects and implementations in new solutions?
10. Do you compare different approaches when creating technical solutions? How we should compare different approaches when deciding final solution?
11. Do you thing we should have referrence architectures? How can you best get information about architectures?
12. How can we make our implementations as component based so that we could utilize re-usability?
13. How do you think we could make our production more effective?
14. What tools we should get and what we should quit using?



## **Appendix 2: Questions in the Interviews**

Interviews were kept using Finnish language, these are translations of the original questions.

1. How do you see the most optimal design process? Which one is emphasized, more agile or more detailed approach (e.g. ADD)?
2. What is the right level or depth of the design?
3. How do we recognize the situation where more detailed and thorough design is required?
4. What should the Technical plan or the design documentation, which is made before actual implementation, include?
5. What level of freedom individual project should have concerning design decisions (used technologies)?
6. How can we best achieve situation where we can make products in the customer specific projects?

