

CHALMERS



Developing a user-centered software development process

- enhancing usability and customer focus.

Master of Science Thesis in the Programme Interaction Design

JOSEFIN NIMSTEDT
MARTIN RODRIGUEZ

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, October 2010

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Developing a user-centered software development process
- enhancing usability and customer focus

JOSEFIN NIMSTEDT
MARTIN RODRIGUEZ

© JOSEFIN NIMSTEDT, October 2010.
© MARTIN RODRIGUEZ, October 2010.

Examiner: OLOF TORGERSSON

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 100

Department of Computer Science and Engineering
Göteborg, Sweden October 2010

Abstract

Competition among IT-companies for market shares increases daily. To win customers' trust, companies must offer high quality and user-focused solutions to an affordable cost. This master thesis focuses on changing the approach of an IT-consulting company, based on an interaction design perspective. Following a self-diagnosis through an analytical study, issues were clarified with a need of focused requirement management. An improvement strategy was applied and through its outcome, a new software development process was created, considering the early involvement of the user. A pilot study was conducted with the created process, emphasizing known methods, streamlining communication and development, in order to evaluate the new approach. It resulted in a customized process that fulfilled the company's criteria, increasing their awareness of a user-centered process' impact. The decision to implement the new development process still remains within the company, since their willingness as organization is a pre-requisite.

Acknowledgements

This master thesis was conducted in cooperation with an IT-consulting company, Västra Götaland region, Sweden, in 2010.

The writers of this report would like to thank all the participants who have taken their time to help us during the master thesis. A special thank to our supervisor and all employees at the IT-consulting company.

We will also thank the supervisor, Olof Torgersson, and the expert consultation from Pontus Engelbrektsson, at Chalmers University of Technology for their support and guidance during this project.

Table of Contents

1. Introduction.....	1
1.1 About the IT-CC	1
2. Background.....	2
2.1 What is a software development process?	2
2.2 What is a user-centered software development process?	2
2.3 Advantages with a user-centered software development process	4
2.4 Disadvantages with a user-centered software development process	4
3. Problem definition.....	5
3.1 Aim.....	5
3.2 Objective.....	5
3.3 Scope	5
4. Methodology.....	6
4.1 An overview of the thesis work	6
4.2 Methods	6
4.2.1 Semi-structured interviews	6
4.2.2 Observation combined with think aloud	6
4.2.3 Content analysis.....	7
4.2.4 Focus group	7
4.2.5 Prioritizing requirements	7
4.2.6 Paper Prototyping.....	7
4.2.7 Persona.....	8
4.2.8 Requirement specification.....	8
4.3 Pre-study.....	8
4.4 Literature review	9
4.4.1 Implementation of user-centered design.....	9
4.4.2 Agile methods	10

4.4.3 Expert consultation	10
4.5 PRE - Products Requirement Engineering	10
4.5.1 Self-diagnosis	11
4.5.2 Improvement strategy.....	11
4.6 Developing an user-centered software development process.....	12
4.7 Pilot study	12
4.7.1 An overview of the pilot study.....	13
4.8 Improving the new user-centered software development process.....	13
5. Results.....	14
5.1 Results of the interviews	14
5.2 Results of the observations	14
5.3 Results of the content analysis.....	15
5.4 Results of the literature review	17
5.5 Results of the self-diagnosis.....	17
5.6 Results of the improvement strategy	18
5.7 The first version of the user-centered software development process	18
5.8 Results of the pilot study	19
5.9 The final version of the user-centered software development process.....	20
6. Discussion.....	21
6.1 Problems encountered	24
6.2 Further advice.....	24
7. Conclusions	25
8. References.....	26
Appendix A – Methods and Guidelines.....	28
Appendix B – PRE - Improvement strategy.....	47
Appendix C - The first version of the user-centered software development process	54
Appendix D - The final version of the user-centered software development process	56
Appendix E - Content analysis	58

1. Introduction

In today's information society, there is a strong need for complex software solutions in business. The actors that provide these services are often IT-consulting companies which are competing for market shares.

Customers' of IT-consulting companies have high expectations of the products, demanding quick deliveries and high quality. In order to get satisfied customers and receive more market shares the IT-consulting companies have to work with clear visions and toward the implementation of appropriate work standards within the organization.

This report was performed in cooperation with an IT-consulting company, hereby undisclosed by name and in the text throughout referred to as the "IT-Consulting Company" (IT-CC) which offers software development services.

1.1 About the IT-CC

The IT-CC is situated, in Västra Götaland and has more than 15 years experience in software development. They are specialized on a particular software application for customized reporting- and analysis tools. They are interested in long-term customer relationships since they aim to work as a backup of both development and support. The range of customers' varies from small local companies to multinational companies as SKF and VOLVO. They have a market place in the boundary between IT-consulting companies and business consulting companies, with the purpose to act as a link between operation close functions and software systems in the organization. The IT-CC strives to have an open-minded service to their customers, where their developing approach is, in general, defined as "develop fast and deliver".

Their projects focus primarily in IT-services and products. A group of 10 people sit in a team-oriented landscape with great knowledge in SQL-server, VB.Net, ASP.net and MS Office. One year ago, the market department lined up the first draft of a software development process.

2. Background

The competition between companies in the IT-world has progressively increased. Customers preferably choose services and products from companies that are affordable, but only when they are assured they receive a fulfilled primary request.

To know what a customer needs is essential but not always easy, especially when the customer does not know that himself. To match the requirements it is important to have a better understanding of the customers' expectations. As a developer it is important to gain the customers' perspective so their needs become clearer. Obtaining a customer oriented perspective helps to earn greater understanding about the customer and in the long run satisfy their needs. This will ultimately generate an increase of market shares.

2.1 What is a software development process?

A software development process (SDP), is a structured management of how a project is planned and performed. The process consists of a series of activities and methods that are shaped and used to create the proposed software (Perez, 2003). Activities, methodology and tools are the fundamental elements to properly obtain a functional process. The activities are used as steps for managing different areas during the project. To successfully achieve a set-up of activities, methods are chosen depending on the correspondence to the specific activity. In turn, specific tools have to be selected in order to ensure activities and methods are correctly implemented.

2.2 What is a user-centered software development process?

In a user-centered SDP, the activities and methodology are focusing on the usability and are oriented toward the users of the service or product. In other words, the actual end-user is the most important factor when establishing this type of software development process. The integration of the usability work and the involvement of the end-users have to be maintained throughout the whole lifecycle, from the initiating state to the final delivery of the system, in order to achieve a result with good outcome (Gulliksen et. al, 2003).

According to Gulliksen and Göransson (2002), a complete process with focus on the end-user consists of four important parts, analyze, design, feedback and evaluation. These are "*analyze*" requirements and user needs, "*design*" usability by prototyping, "*feedback*" for planning the next iteration and lastly, to "*evaluate*" use in context. The process is iterative and contains active user involvement during the whole implementation. To fulfill these four parts of the process, user-centered methods are followed to get the best understanding of the users and their needs. The process should be based on the key principles specified below.

User focus and user involvement

The users' requirements, goals and situation should guide the whole development and organization, with the user being a central part of the development. It is important to prioritize the best for the users, instead of developing the most effective technical solution. The users should be involved early in the development and tight communication should be a representative part during the whole SDP. It is important to recognize the difference between the user and the customer. The user is the individual using the system, while the customer is the client. Another important aspect is to identify parts that should be implemented in the process, such as analysis, design and evaluation (Gulliksen et. al., 2003).

Evolutionary development

The system should be implemented with iterations together with the users. It should contain an analysis of the users' requirements and situation, as well as a proper design and documentation of the evaluation with items that can be changed or implemented (Gulliksen et. al., 2003).

Collective and shared understanding

Requirements and design suggestions should be explained in such a way that all parties can understand. The user should get a clear view and understanding of the system (Gulliksen et. al., 2003).

Prototyping

Prototyping should be an early and continuous element in the process. Simple sketches and prototypes are to be used to visualize ideas and solutions and to support the iterative process (Gulliksen et. al., 2003).

Evaluation

The process should be controlled by measureable goals and use in context. The design has to be evaluated for capacity to fulfill set goals and usability criteria. Testing of the design has to be performed early to get a view of the users' thoughts and feelings for the design (Gulliksen et. al., 2003).

Design activities

The process should contain activities focused on the user and dedicated design methods. One should implement methods that create awareness of what is best for the user. The result should be a well structured and conscious action (Gulliksen et. al., 2003).

Interdisciplinary teams

A professional attitude performed by teams with different skills and expertise should be a central part of the development. Different competences contribute to the final solution, where they should work close together and interdisciplinary to cover all parts of the development process (Gulliksen et. al., 2003).

Integrated system design

All components with an impact on usability should be integrated. They should be implemented in parallel, in interdependence (Gulliksen et. al., 2003).

Usability experts

Usability experts should be involved early in the process to support and guide the development team according to the users' situation and needs (Gulliksen et. al., 2003).

Customize processes

The process should be specified and adapted to fit the organization based on its needs. The organization has to take responsibility of how to work with usability. It needs to specify what activities should be a part of the SDP, as well as to see that they continuously improve and change so that the approach fits to any kind of project (Gulliksen et. al., 2003).

A user-centered attitude

Establish a user-centered attitude and educate the teams and stakeholders regarding the importance of usability- and user-involvement. All developers should meet the user, preferably through an incepting meeting in the user's environment. Every member of the project team has to become aware of the concept of usability, depending on their role in the project (Gulliksen et. al., 2003).

2.3 Advantages with a user-centered software development process

There are several advantages of using a user-centered design and implementing it to a SDP:

Improved customer satisfaction

According to Sharp (2007), if the users are involved during the process they will feel a stronger bound to the product or service creating a better reception at delivery. When the user's needs are accomplished the product creates better user experience and result in an improved user satisfaction (Tec-Ed, 2009).

Reduced development costs

User focus and iterations during the process will give a better understanding of their needs and how they work. It will speed up decisions and prevents errors in later phases, which need re-design. At the same time it may reduce costs for support and learning of the systems (Tec-Ed, 2009).

Better definition of product requirements

A focus on the end-users clarifies their needs and desires, with an obvious advantage if identified at early phases. It further provides a better understanding for current systems and of the users' situation. It prevents developers from overlooking and misinterpreting requirements that are unstated (Tec-Ed, 2009).

Future sale and implementation

User-centered designed activities contribute to exploring features and usability issues that can be addressed in further releases. It is possible to find new and unpredicted patterns of product use, and needs for new or related products and improvements for future sale (Tec-Ed, 2009).

Provides a competitive advantage

If one can meet customers' needs with superior user experience and outstanding usability, by offering mature and future features, it will help to distinguish the process from other competitive companies. This will provide a competitive advantage on the market (Tec-Ed, 2009).

2.4 Disadvantages with a user-centered software development process

Where there are benefits there are also disadvantages. It is also true for the user-centered approach. Despite that it may be preferable, some aspects must be considered. The user-centered approach is depended of the end-user that must have an active participation during the whole development process, to ensure that quality is preserved and enhanced. In some cases, several stakeholders have to be involved, which may adversely affect the development by causing delays (Abrams et. al., 2004). Problems that may arise in connection with delays often increase the running costs. As unforeseen expenses are never desirable, user-centered aspects are not prioritized and therefore lose its advantage.

3. Problem definition

The IT-CC recently decided to standardize their working routines in a business and SDP, to improve their consult services and products to continue offering high quality work. Their goal was to give project leaders a good overview of the different projects' status and progress, to keep uniform standard of their projects and to maintain customer satisfaction. Despite efforts, the standardization has not yet been accomplished since the organization still maintains developers working after old routines and experiences.

3.1 Aim

The aim of this thesis is to analyze the IT-CC's working routines and improve their SDP. It ultimately aims to enhance the usability and customer focus of an user-centered SDP, emphasizing known methods in the interaction design area.

3.2 Objective

The objective is to contribute to an improvement of the IT-CC's SDP and usability approach. Practicing this knowledge, improvements in their business- and development process ought to be used in future projects, hopefully resulting in higher customer satisfaction, better graphical user interfaces and overall usability.

The major concerned area is the design process. The major approach is a change in methodology and, in general, a devised strategy to determine how developers think and act in regards to usability and user-oriented perspectives.

3.3 Scope

The master thesis focuses on changing the IT-CC's way of thinking from an interaction design perspective. It will deal with the interaction between the IT-CC, their customers and with the end-user. Since the thesis is part of the Master of Science Interaction Design Programme, it will only focus on the SDP and not the initiating sales process. The IT-CC will develop it themselves. Moreover, the development process is to focus on which methods are to be most appropriate to manage requirements and to perform usability testing.

When developing the SDP, we had in mind to use several methods used in the interaction design area. Some of these methods were never implemented in the final version of the process. For that reason we have omitted the unused methods in the report.

4. Methodology

In order to perform the master thesis, an appropriate methodology approach was defined. The methodology included several methods with aim to facilitate the process of analyzing the IT-CC's working routines with customer focus. Methods were also used to be able to reform their SDP into a more user-centered approach. In this section, a description of the methodology and the used methods is presented.

4.1 An overview of the thesis work

The thesis is divided into seven steps where the initial step concerned the scoping and planning of the study. This step was done in agreement with the supervisor at the IT-CC to state the terms of the study plan. In the second step a data gathering was performed at the IT-CC and also with some of their customers. We learned about their approach when designing systems and how they communicate with each other.

The third step of the thesis included a literature review where relevant information was collected for analyzing the gathered data and finding the best practice user-centered processes for projects that are similar to the ones at the IT-CC.

In the next coming step the gathered data is analyzed and discussed to find the IT-CC's issues regarding communication with the customer and interface design. The results are presented in the next step where a new user-centered SDP is described to meet customers' needs, focusing on reaching greater customer satisfaction. In this step an improvement strategy was devised as a future guide for the IT-CC when implementing the process into their organization.

In the sixth step, the process for a user-centered approach was tested in a pilot study. The communication with the customer was closer and the developed process tested to see which parts worked properly and which did not. In the final step, the process was evaluated according to the results in the pilot study step. Necessary changes and improvements are ultimately proposed to the IT-CC's organization and to their projects, in terms of efficiency.

4.2 Methods

The master thesis included several methods. They were used during the performance of the pre-study and the pilot study. They are explained below.

4.2.1 Semi-structured interviews

A semi-structured interview is a data collection method that is used to get a better understanding of a customer. It is based on a number of carefully selected questions to be addressed. Beyond the specific questions, there is also room for further investigation that may be done during the interview (Bernard, 2002).

4.2.2 Observation combined with think aloud

Observation is a data collection method based on observing and documenting how a user interacts with a system. An observation can be done in several ways, but the most traditional is that the observer sits in the background and documents every step of the user's interaction with the system. There are benefits with observations, you get better understanding how the user interacts with the system, the interactions are observed in real time and reality emerges. Details

that would normally be discarded are documented as it happens in this very moment. This also includes details that are reformed in time, when our memory fails us (Kaulio et. al., 1999).

When performing an observation a number of scenarios are created that the user shall carry out. An observation can be held in the user's natural environment. The observations should be documented or recorded to include details. The observation also allows gathering the user's reaction, with help of the think-aloud. Think aloud allows and encourages the user to speak freely and give feedback at the same there is an interaction with the system. It also provides the opportunity to express feelings and thoughts (Van Someren et. al., 1994).

4.2.3 Content analysis

Content analysis is a method used to compile large amounts of data from different data-collection methods. Data may come from different types of collections, such as documentation or recorded media. With help of the method, data are sorted and their meaning is presented and clarified.

An important consideration when working with content analysis is to respect all data, regardless of their content. All documented data are written down in a joint document. All text is divided into segments and treated separately, this part is also known as "meaning units". All meaning units are read through several times to gain a deeper understanding of their actual meaning. Later on, they are iterated and cut down to more descriptive segments called "condensed meaning units". This is performed for pitching out the important parts of the text. From the updated segments an analytical process is initiated, whose task is to reveal an interpretation of the underlying meaning in all the different segments. Similarities between the segments become obvious and are divided into common subthemes. As soon as all subthemes are revealed, an overall common theme for the whole content analysis is presented (Graneheim & Lundman, 2003).

4.2.4 Focus group

Focus group is a method used for retrieving qualitative feedback from users or persons involved in a software development project. The fundamental use of focus groups is that all participants are gathered, speaking freely about the software being developed. The discussions can be about views, thoughts and ideas of the software. The fundamental idea is to try creating a discussion with all participants, receiving feedback. Based on the received feedback, it is possible to retrieve new ideas for updates in the development plan (Nielsen, 1993).

4.2.5 Prioritizing requirements

Prioritizing requirements is a useful method helping the development team to know what should be developed first according to the customers' requirements. There are several different approaches for prioritizing requirements. The found requirements need to be compared both from the development team and the customer as the perspective normally is not the same. The customer prioritize the requirements according to his needs, afterwards the development team do the same out of their perspective. The priorities are compared to find similarities and discuss further for an agreement what should be prioritized first (Kaulio et. al., 1999).

4.2.6 Paper Prototyping

Paper prototyping is a method used during the design phase of a SDP. It helps developers at an early stage visualize for the user, how the applications' interface and navigation can work in a

final solution. Not only that it helps the developers', it also gives an opportunity for the customer to understand how their desired application starts to take form, giving them an chance to see and influence how their application will look like. Nothing is yet written in stone, which allows the user to express their views on the prototype, giving them the opportunity to reshape it without negative consequences. Paper prototyping is based on sketches with pen and paper. The sketches may well be scenario-based, giving an opportunity to enable navigation in the prototype as well. The prototype is developed with the requirements and functionality in mind. Several sketches should be performed so different alternatives are realized (Osman et. al., 2009).

4.2.7 Persona

Persona is a method used to get a clear picture of different user types. The persona is only fictional and may appear in some cases as a stereotype. The reason is for really finding differences in between the different user types, making it obvious how the different groups behave from each other. A specific persona describes what are its group's specific needs, behavior and experience. With help of this knowledge it is easier for the developers to filtrate and see what specifically is important for a specific user (Kaulio et. al., 1999).

4.2.8 Requirement specification

Requirement specification is a method that collects on a single sheet, all the functionality to be included in the application to be developed. The sheet is divided into two parts which distinguishes the functional requirements and non-functional requirements. Functional requirements are based on the systems behavior it must perform, while non-functional requirements are requirements that present the attributes of what the system consist of (Andersson & Nordgren, 2010).

4.3 Pre-study

During the data gathering phase, important data were to be retrieved. Since the result from this stage will work as a foundation for the final result, it demands precision in order to succeed with the analysis further on. Qualitative data sampling is used in order to achieve good quality. There were no possibilities to reach a large amount of users and therefore quantitative data could not be gathered. For the qualitative approach, interviews and observations were done with the IT-CC's developers, project leaders and with users of one of the IT-CC's customers. The results were summarized in a content analysis to receive an overview of the gathered data.

Seven interviews were carried out with the employees and six interviews were also conducted with already existing customers. All interviews were carried out by the thesis authors, cooperating. Both had their own specific tasks to take care of to ensure improving the method for collecting data. One person was holding the actual interview, with direct focus on the interviewee. The other person was responsible to document everything, as literally as possible.

In order to get a picture of how the communication with the user has been during earlier projects observations were performed at one of the customer's office. Six users were observed on a delivered application, developed by the IT-CC. The application was a time-report system, displaying a calendar. Every employee at the company was supposed to login to their account and time-report by adding their worked time in the application. The observations were based on six tasks to perform, where every move and action was recorded and documented for further analysis.

4.4 Literature review

As mentioned in section 4.1, the third step of this project concerned a literature review. The purpose of the literature review was to study, highlight and update what is known in the area of user-centered development and software development processes, in order to obtain guidelines and frameworks that can be applied throughout the project. The review focuses on both best practices and approaches that were not suited for the thesis purpose. We have used sources that are academically-oriented, but will focus the results on applied theory rather than academically theory.

4.4.1 Implementation of user-centered design

The expression “user-centered design” can be experienced as vague and many authors have described it with different approaches. One of the definitions written by Preece et. al. (1994, p. 772), defines “user-centered design” as:

“...an approach which views knowledge about users and their involvement in the design process as a central concern.”

Norman (1986) claimed the following:

“But user centered design emphasizes the purpose of the system is to serve the user, not to use a specific technology, not to be an elegant piece of programming. The needs of the users should dominate the design of the interface and the needs of the interface should dominate the rest of the system.”

Despite the presence of different point of views in the literature, these approaches are a central part in developing systems that should be usable.

Human-centered design processes for interactive systems

According to ISO 13407 there are four important steps to follow the definition of human-centered design. Firstly, the user should be actively involved in the development of the system and developers should have a clear understanding of both users and system requirements. To follow the ISO standard an appropriate allocation of functions between users and technology should be implemented together with iterations, multidisciplinary design and implementation. The ISO 13407 is not a complete SDP; it is an approach that can be implemented together with other methods and processes Gulliksen & Göransson (2002).

Implementing a user-centered system development process

Gulliksen & Göransson (2002) described an approach for implementing a new SDP in an organization. The approach consists of seven steps to be followed. The first one is to evaluate and describe the existing process, together with advantages and consequences. Interviews with involved persons should be a part of this step. After the first stage, a specification of the different roles of the organization should be done. One should specify what kind of experiences and competences the employees have, with focus on interfaces and usability. In the third step, the organization should present their definition of a user-centered design process. The organization should take an active stand to what key questions in user-centered development are present. The organization should also specify the key areas where the organization needs to change and focus their future development activities. In the fifth stage, new methods should be identified and implemented, focusing on users so that the organization further on can identify methods

that are not useful and reduce them. In the last and seventh stage of the implementation, the new process should be adapted and specified for every specific project.

4.4.2 Agile methods

Agile methods are known as good approaches together with user focus and interaction design, because of their short iterations and the closeness of communication with the customer (Sharp et al., 2007, p. 454). Some are considered particularly useful in this project and are described below.

Dynamic Systems Development Method

The Dynamic Systems Development Method (DSDM) consists of nine principles to be applied in a project. The DSDM is about delivering results fast and cheap with focus on the user. The nine principles to follow are: (i) the involvement of the active user, (ii) teams must be empowered to make decisions, (iii) testing is integrated throughout the development, (iv) fitness for business purpose, (v) iterative development, (vi) incremental development, (vii) all changes are reversible, (viii) requirements are of high level and (ix) a collaborative approach between stakeholders. In DSDM, prototyping is more important than planning and, therefore, best suited for user-focused projects (Stapleton, 1997).

Rational Unified Process

Rational Unified Process (RUP) is owned and developed by IBM Rational Software and is based on six different best practice approaches. These are: (a) develop software iteratively, (b) manage requirements, (c) use component-based architectures, (d) visually model software, (e) verify software quality and (f) control changes to software. The process is suitable and adjustable for every organization and configurable for different situations (Rational the software development company, 1998).

4.4.3 Expert consultation

While the literature studies were conducted, we have consulted an expert, Pontus Engelbrektsson, from the Department of design and human factors, Chalmers University of Technology, regarding our different approaches of work, owing to his experience. We were recommended to take a closer look at a book that he was sure it would fit us well, using the approach we were doing. The book that he recommended is PRE, Products Requirement Engineering, Customer Understanding in Product Development (Kaulio et. al., 1999). It can be used as a guide for companies who are interested to improve their customer understanding when developing products or services.

4.5 PRE - Products Requirement Engineering

Products Requirement Engineering (PRE) is a strategy used for companies to be able to measure and improve their degree of maturity, regarding process development and customer management (Kaulio et. al., 1999). PRE focuses principally in three areas; process, manning and methodology. To measure the degree of maturity of the company, a self-diagnosis is performed. From the measurement, a strategy for the company can be devised and used to retrieve an approach on how to improve these areas and rise the degree of maturity.

4.5.1 Self-diagnosis

Self-diagnosis measures the current maturity level of a company and shows which parts that need attention to enhance their future work (Kaulio et. al., 1999). Self-diagnosis is divided into three areas, process for customer management, manning of projects and methods for customer management. All three areas are related to a scale, operating from one to four, see matrix below.

IT-CC	Process	Manning	Methods	
Level 4				Constantly improving Implemented Defined Sporadic
Level 3				
Level 2				
Level 1				

Figure 1. Matrix representing areas and maturity level

The different levels are defined as following:

Level 1, also known as the sporadic level, is the level where no rules are defined on how situations should be resolved and who is going to resolve them. Much of the responsibility is being handed over to individuals and it is upon their reasoning that the project eventually moves forward.

Level 2, the defined level, shows that the company already knows how to deal with upcoming situations as well as how to deal with customer requirements. This type of approach has at its current state already been used in a few projects.

Level 3, the implemented level, shows that the company has reached the step where the defined level is used as a standard by the organization in all its projects.

Level 4 is considered achieved when the defined and implemented level is being fully used, but also when there are constant improvement and development ideas for future work.

To estimate the maturity of an area, an assessment is performed. The assessment is measured with help of specific criteria that the company needs to master to be able to reach an upper level. If one area does not manage to meet up to all criteria's in a level, the area remains on the level immediately below.

A self-diagnosis of the IT-CC was carried out using documentation from the content analysis. The content analysis contained data from interviews and observations that had been performed at the IT-CC and their customers.

4.5.2 Improvement strategy

As soon as a self-diagnosis has been carried out, the criteria that have not yet been fulfilled are presented. The next step for achieving higher level of maturity is to follow an improvement strategy, based on the self-diagnosis results and which is followed in a predefined order (Kaulio

et. al., 1999). The order is adjusted because when a specific area has been completed, it will help the next areas to complete their upcoming levels, as depicted in the matrix below.

IT-CC	Process	Manning	Methods	
Level 4	8	7	9	Constantly improving Implemented Defined Sporadic
Level 3	6	4	5	
Level 2	1	2	3	
Level 1				

Figure 2. Matrix presenting the pre-defined order of improvements

The company may be on different levels in different areas. The first step is always to make sure that all three areas are in balance in level. As soon as balance is achieved, the predefined order is followed.

From the product requirements engineering improvement strategy, we limited the thesis only to help the IT-CC raise the process maturity. This was the first step to achieve balance in all three areas. The remaining improvements are handed over to the IT-CC to manage themselves.

4.6 Developing an user-centered software development process

As the IT-CC already had a pre-defined SDP, we created a process based on their current basis. We primarily wanted the process we were creating to have a more user-centered perspective, at the same time that the IT-CC would recognize and be able to apply to it easier. When searching for appropriate methods, we had in mind that the methods were supposed to be applied within the limits of what the IT-CC would be capable to use in reality. The methods we focused on were to be used throughout the whole SDP and would be as time-efficient as possible. The idea was that they could be used in connection with meetings or other mandatory occasions. The different methods had to include practices requiring management, design and evaluation testing.

To facilitate the adaption of the new process, a user guide, called Methods & Guidelines (see Appendix A), with all implemented methods was created for the IT-CC. The user guide will work as an aid for the developers together with the process until the methods are standardized and incorporated in the organization. The guide will also help new employees to adapt easier to the IT-CC's working routines.

4.7 Pilot study

A pilot study aiming to test the feasibility of introducing the proposed user-centered SDP into the organization was issued. The pilot study was conducted as a live project at one of the IT-CC's customers. It was supposed to include all the important methods and parts of the created user-centered SDP. The project involved developing a validation tool as an aid for product managers at an electrical wholesale enterprise. The project was conducted in communication with the development team and the project manager, working as usability consultants. The work included customer visits, user-interface design and guiding the developers towards a result with high user-focus. Methods and Guidelines are used according to the created user-centered SDP.

4.7.1 An overview of the pilot study

The study began with an initial meeting with the project manager, explaining what the project was about. Already in this stage some few paper prototypes (see section 4.2.6) were created. At this initial meeting in the customer's office, we carried out semi-structured interviews (see section 4.2.1) with users, together with an open meeting about more technical details. The paper prototypes were shown and discussed with the users.

Back at the office, the results were compiled and presented in a content analysis (see section 4.2.3) and they were discussed with the project leader. Personas (see section 4.2.7) were created to convey the user's needs and wishes and their experiences to the developers. An internal introductory meeting was also conducted to start the project within the organization, with responsibilities being divided between project members. The requirements from the content analysis were summarized in a requirements specification (see section 4.2.8) and prioritized (see section 4.2.5) by both customers and the project team.

Further, the user interface was developed based on the paper prototypes and the users' feedback from the introductory meeting with the customer. A draft was presented in the upcoming meeting with all users. A focus group (see section 4.2.4) was held, including all users discussing the graphical interface and additional functions. Observations were performed during the meeting, with each of the users. The method think aloud (see section 4.2.2) was used.

The result from the performed usability test was summarized in an updated requirements specification, presented to the clients for further sale and implementation.

4.8 Improving the new user-centered software development process

In order to improve the first version of the SDP, the evaluated process was updated to fit the IT-CC as an organization. Based on the results of the pilot study, a re-structure of the process was performed. The re-structure included removals of methods and activities, but also additions of certain activities.

5. Results

In this section, the pre-study results are presented, including the outcome from the interviews and of the observations, alongside interpretations. The results from the content analysis and from the pilot study contributed to the final version of the user-centered SDP.

5.1 Results of the interviews

When interviewing, a lot of information was retrieved, that could be put into good use. From the interviews, connections and similarities could be found directly. All the information was also used later on for the content analysis and can be useable for further research. Important quotes have been highlighted below because we see them of particular importance;

“People are not following the development process at the moment. People are used to work with their own routines. I’ve started looking at it. The process will most certain be a requirement in the future.”

Based on the interviews conducted at the IT-CC, some common situations arose. A frequent similarity was that the staff is used to work after routines and their own experiences. Another similarity was that they did not follow nor used the SDP. Some persons were not aware that they should follow the existing SDP.

“The application doesn’t seem to work properly in Firefox.”

During an interview with a client, we became aware of a situation that had arisen. The IT-CC had delivered a web application that only worked properly with Internet Explorer, but the client was not aware of this restriction. In their everyday work, the employees at the customer’s work in different web browsers, even different operating systems. This caused frustration, as the employees had to switch browsers or borrow colleagues’ computers to be able to use the application.

“It took a long time before we got to see the web interfaces. In January, a month before delivery, we saw it for the first time. Before that we haven’t seen anything, nor sketches or prototypes. I would have liked to have seen it in an earlier phase.”

As the quote tells us, the client actually never saw how the prototype of the application started to take form. The interface could have been shown earlier as there was time for that, but in this case, that was not included in the planning. By the time it was firstly seen, it was already too late to comment on own opinions and thoughts of the design and navigation of the application.

Other important points to mention are misunderstandings that arise too often. From the clients’ perspective, the uncertainty is not sustainable as the customer is not prepared for what is there to come. From these problems, faults become sequenced as the communication is deficient and overall gives no quality to the process.

5.2 Results of the observations

Observations were conducted on six employees doing four pre-defined scenarios in the time-report system. The observation was combined with think aloud to obtain additional information, information that was used later in the content analysis for further research.

The observation clearly showed that various employees had their own way for navigating in the interface, as one of the tasks showed:

- The task was to navigate to a specific week in the time-report application. Some of the employees used an available shortcut to move directly to the specific week while others went gradually through all weeks. Both ways solved the task, but gave different output from the user. There were different reasons for this, including the fact that some of the users did actually not know that a shortcut existed. Other users found it more convenient to just click through all weeks. Using the shortcut was by far the fastest way to solve the task and it also disabled a popup that occurs if stepping through the weeks. An irritation was clear for the employees that did the stepping as the pop-up showed up for every single week.

Developers mentioned to us earlier that the application was never created for stepping through several weeks, even though it was possible to do so. The correct way of doing this, avoiding the pop-up is by using the shortcut. Still, the problem was there, still some users did not know that the shortcut existed.

Another interesting aspect from the observation was that the time report was developed for daily reporting. This is not the case at the customer's. The customer time report differently, sometimes weekly or even more spread, as monthly.

- Another task was to time report a specific date in the future. Every single user confronted the same problem; they needed to count through the days to be sure that they were time reporting correctly. All users knew how to solve the issue by just counting, but at the same time they found it unnecessary, as it would be more convenient to have the date stated for every specific day, which it was not at the moment.

The result of the observations did show that the IT-CC had insufficient communication with the users. There seemed to be several misunderstandings on how the system was suppose to work in the end. The IT-CC's communication with the users must be improved in order to perform better requirement management elicitation.

5.3 Results of the content analysis

The results of the content analysis are a compilation of all the documentation retrieved from the internal and external interviews and also data from the observations, see Appendix E. Our interpretations resulted in 13 different sub-themes. All sub-themes bottom in two key issues, managing requirements and follow-up. The lack of methodology and poor understanding of the customer created unsustainable scenarios, which lead to unjustified decisions and uncertainty during development. The sub-themes are presented below with a solution on the specific problem. Some of the sub-themes have been assembled because they are based on the same issues.

Difficulties in managing unspoken expectations

Customers have expectations on their wanted systems. In some cases the expectations may be so obvious that they are not explicitly stated. If the expectations remains implied, this may increase the risk for dissatisfaction and misinterpretation. At this moment, these situations are not taken care of. They have to be considered to avoid the risks they may cause.

Inadequate internal communication

During development, it is important to have a general understanding of the customer. Communicating through several channels may interfere with the primary information, reforming it to something different from the original. The IT-CC still has vague internal communication that could be improved.

Inadequate external communication

When external communication is not good enough, important aspects may be lost. Deficient communication increases the risk of claims being vague and uncertain. Customers do not always know what can be developed technologically and if this is not clarified, opportunities are limited for additional sales.

Undefined process

The IT-CC's process approach is perceived as sporadic. The development seems inconsistent and is more focused on completing tasks rather than making the applications useful for the customer. A more consistent approach is needed making sure nothing is missed out and implemented in a correct order.

Inadequate in design phase and interface management

There are no standardized methods used for designing interfaces and its navigation. The IT-CC needs to start making use of tools and methods motivated for design decisions. Follow-up is needed as the interfaces might need re-designs and updates depending on the customers' needs.

Unjustified decisions

Far too often, decisions are made without any defined criteria. No methods are used to compare different important decisions. Customers often have too strong impact when deciding and, therefore, no proposed alternatives are presented from the developers. Developers need to feel comfortable when presenting alternative ways of solving problems.

Inadequate customer knowledge

The IT-CC's existing SDP is not focusing on the end-user. Knowledge about their customer is vague, which results in applications not fully adapted to the user. If there are different types of users with different needs, it may result to unfitting solutions. The understanding of the customer must be improved, removing this type of issues.

Inadequate in functionality management and methodology for managing requirement and follow-up

The IT-CC's customer management consists of defects, which affect the functionality during development. The issue is that there is no methodology to elicit and manage requirements. Unknown requirements may be missed out, which result in lost functionality and poor usability. The IT-CC needs to implement proper methodology, making sure that the requirements are correctly managed.

Losing potential additions

The content analysis shows that the IT-CC does not capture further requirements after delivery and do not provide their customer with significant additional functions or features. Having a standardized and careful follow-up provides a stronger customer focus.

5.4 Results of the literature review

During the literature study, many approaches and processes were studied in order to find guidelines during the project, attempting a user-centered SDP.

The findings from the literature review showed that there are approaches suitable for our purpose with the project. A useful approach is the one presented by Gulliken and Göransson (2002), (see section 4.4.1), about implementing a new process into an organization.

Some established agile SDP were studied that could be suitable for the purpose of the project. Because of the IT-CC's approach in projects and the construction of the organization, the choice was not to implement these.

5.5 Results of the self-diagnosis

As mentioned earlier, a self-diagnosis was performed to determine which maturity level the IT-CC had achieved at present. Based on all performed interviews and observations, we analyzed and compiled a matrix showing this result:

IT-CC	Process	Manning	Methods	
Level 4				Constantly improving Implemented Defined Sporadic
Level 3				
Level 2				
Level 1				

Figure 3. Matrix of the IT-CC's self-diagnosis

The matrix shows that the IT-CC had reached level one in process, and level two in manning and methods. The results may be perceived as disturbing, which is not the case. The result barely indicates that there are details in the lower levels that are still missing for maturity to increase. As soon as these details are considered and resolved, a re-analysis should take place.

Assessment of maturity level during the process

The self-diagnosis analysis did show, that in their current situation, a SDP is partially handling customer requirement. Still, it remains that the company and the employees begin to use it and, further, that it is included in their everyday work. The company is not achieving the second level in the process at this moment and therefore stays at level one.

Assessment of maturity level in the manning

Based on analysis of manning maturation, it is clear that there is an inadequate communication between the marketing department and the development department. During the project, not all developers were in direct contact with the customers, yet there were still mediators communicating with all the people.

Assessment of maturity level in the methods

The company is only partially working with defined methodology, pushed by individuals. Knowledge and experience of method management is still generally limited and

there is no predetermined selection of which methods to be used when developing, thus reaching level two.

5.6 Results of the improvement strategy

With help of the results of the self-diagnosis, an improvement strategy could be realized. As the thesis focus was to implement a new SDP and the improvement strategy could give more results than necessary, we decided to limit the strategy to the segments we were going to work with.

The first step to solve is to make sure that the company gains balance in level. In this case the process was the first target as it remained on level 1, see figure 4.

IT-CC	Process	Manning	Methods	
Level 4	6	5	7	Constantly improving Implemented Defined Sporadic
Level 3	4	2	3	
Level 2	1			
Level 1				

Figure 4. Matrix presenting the IT-CC's improvement strategy in predefined order

The company has partially reached level two, but there were yet criteria not reached. A SDP has been acquired. However, it is yet not complete regarding requirements management and, further employees were not familiar with it. To achieve level two in process maturity, the company has to define their process, determining how they should handle customer requirements. In addition, few of their projects must follow this process.

The next step to be followed by the company is to identify those phases, decision points and activities for the new process and also, try to anchor them into the organization. Key activities and methods have to be identified. Activities have to be gathered in to the main steps of the developed process. Also, the new process must be recognized by all the employees, with a common vision being shared. In order to ease the implementation of the improvement strategy, an improvement plan was created for the IT-CC, see Appendix B. The improvement plan would facilitate the implementation for the IT-CC. Not to forget, employees' thoughts and opinions are most relevant. They are the ones who shall work with the devised process. As soon as all these elements are under relative control, the company will reach balance in all three areas, and can move onto continued maturation.

5.7 The first version of the user-centered software development process

According to the results from the content analysis in the pre-study, the process will be developed to solve the presented issues. It will be implemented with methods which are relevant. The choice of methods is based on experiences, but also methods known as time-efficient with good quality found in the literature.

In order for the company to reach level two in maturity, the first step is to standardize a SDP. It should be implemented with the purpose to improve the elicitation and management of requirements finding main steps, decision points, and activities and in the end anchoring them

to the IT-CC. The process is designed to be performed with an iterative approach and with customer focus, keeping continuous customer communication. The user-centered SDP is presented in Appendix C, giving a descriptive overview.

The first step is “Sale”, including the activity to present the IT-CC and how they work with the customer. It is important that the sale’s manager focuses on informing the customer about the IT-CC’s focus on usability and user involvement. The purpose with this step is to make the customer to understand about the careful pre-study and their necessary involvement from the end-users during the whole project.

In the following step, “Project initiation”, the project manager is appointed. The project manager’s assignment is to judge whether the projects is feasible according to time and budget proposal. If this is the case, then the following activities handle all the important areas that consider customer knowledge and understanding. During this initiation phase an interview is included. The interview is combined with a structured protocol, allowing open questions and further discussions.

With the collected information from the meeting, obtained data must be analyzed. The data result in requirements that are presented and prioritized. The reason for retrieving relevant data is to make it easier to interpret and manage. Methods are used to identify different user types, finding requirements and the understanding the end-users need. Prioritizing the obtained requirements helps the developers to know what to focus on. With help of the structured information the next stage commences, “Project planning”.

The purpose of the “Project planning” is to man the project, present an initial time plan and create a general understanding. The appointed members obtain relevant information regarding the project and they are assigned to different responsibilities.

The next step is “Design”, where sketches and flows are created. The chosen methods are used to see the differences between the user-types and flows when interacting with the system. The visualization of the user interface is tested, investigating if the usability is acceptable. As soon as the design for a specific prototype has been chosen, the “Implementation” phase begins.

The “Implementation” phase is the programmers’ phase, where the application starts to take form in the digital space. While the application takes form the programmers make sure that all the requirements are fulfilled and working properly.

As soon as the application is complete, a delivery to the customer is progressed. The IT-CC installs and guides the customer on how the application is working, keeping a service-minded approach. The project is now finished, but the process work continues. The follow-up is a procedure that will take place as long as the system is in use and there are functions that can be implemented in new projects.

5.8 Results of the pilot study

When testing the developed user-centered SDP in the pilot study, conclusions about the first version’s relevance came to our attention. Both time and budget were limited, which caused the some activities in the created user-centered SDP were unable to be performed.

Because of these restrictions, the prototyping was decided to be performed earlier than planned. This proved to be an advantage in the development because prototypes could be presented for users at an earlier stage.

The time budget was a significant factor for the company as they provide an immediate service to the customer. The time from the initiating meeting until the delivery of the application was only two and a half weeks. During this time all types of tests and methods should be performed according to the created process. It was simply not enough time to perform everything as it was planned from the beginning.

Another factor we had to take into account was that the IT-CC had to make decisions about functions and features found during the interviews and observations. Depending on how the proposal to the customer looked like, the IT-CC took different approaches deciding what would be implemented or not. The IT-CC proposed the customer this functionality and features as additional sale, but the customer denied some of the functions.

5.9 The final version of the user-centered software development process

The changes made in the user-centered SDP are done in order to improve the result further and to fit the process into the IT-CC's organization and the kind of projects they are facing. Some of the activities were removed, replaced by more appropriate methods, while some were just reformed. In projects like the one in the pilot study, some of the methods were superfluous and, in consequence ought to be also removed. The changes and improvements in the first version of the SDP resulted in the final version, see Appendix D.

"Sale" was moved to a process of its own for the IT-CC. This part should be extended and developed by the IT-CC. It was realized that sale and marketing did not belong to the actual SDP.

In the next phase, "project initiating" and "project planning" were joined as one phase, since they did not have a clear, meaningful transition. The manning activity was moved to an earlier phase with the purpose to put more focus on developers meeting with the customer. This would improve the project members' general understanding of the end-users. As this is so important, an internal introductory meeting was also implemented.

Prototyping should be a central part of the development. In the pilot study the initial design of the application were sketched before meeting the customer. This contributed to a discussion about the user interface at an initial stage and provided the user with an opportunity for input to the development.

6. Discussion

It is essential to understand what the customer needs to be able to develop new products and services. It is of great importance to improve the communication flow between the different people involved in the projects, a subject that has proven not to be so easy. The IT-CC's task is to facilitate the use of software and meet customers' wishes and needs. However, this is not always the case as developers still lack in using a defined process with user-focus, staying in old routines and experiences. As an established IT-company, why do they lack a software development process? From our point of view we see the benefits a defined process can give a company that provides services. The reason can be that the company is too small to see what they can earn from using one. Another reason, also depending on the company's size, can be that they do not have a department specifically processing these types of questions.

This thesis has analyzed the way a relevant IT-consulting company works in different projects. The reason was to develop an improvement strategy for the company to implement. The goal was to provide views to enhance usability from a user-oriented perspective, emphasizing known methods in the interaction design area. As we already mentioned, we can see benefits with a user-centered perspective. Because of our knowledge of methodology and customer focus we felt that we could provide this to them. But the question is, will they seize the competence we have shared? As we only have shared this competence to a part of the organization it is important that it reaches the rest of the company. This task is not a part of our project and it is important that they take this forward; otherwise our impact will not be of any significance. It has come to our attention that they already have an employee possessing the competence in the interaction design area. Unfortunately they do not use him in this area.

The thesis has been built following a planned set up, with combined input from the university supervisor and the supervisor at the IT-CC. The supervisor at the IT-CC was essential for us, to understand the needs and purposes of the company. We also reached an agreement for the procedures to be undertaken while developing the thesis project, confirming a proper study plan. The project manager at the IT-CC has been very important for our implementation, as he is the one with great opportunity to share further information to the employees. In addition, he is the link between developers and the marketing department. The opportunity we had to meet an expert in the field also contributed to another perspective, because he presented the facts of a real project and how time and budget matters. This is a big difference comparing with the ordinary school studies we are used to. These two factors are very crucial for the outcome of what the process would look like.

Based on the collected data and the following analysis, a self-diagnosis was performed. The results of the IT-CC's self-diagnosis, aimed to determine the level of maturity it had achieved at present and based on interviews and observation, it showed that they had reached level one in process, and level two in manning and methods. The results indicated details at the lower levels are yet missing for maturity, to enable the IT-CC to increase. The IT-CC was missing partial handling of the mentioned requirements by end-users, but still places it at level one of maturity. This was identified as a major issue, and the team members of the IT-CC should need to change some of their routines in their everyday work. We think the reason for the low level of maturity depends on that the company has relied on individuals and because of that work in their own way independently according to their skills. Raising maturity level is nothing that the individuals

can take care of themselves, the decisions must come from higher authorities, and there is a need of a conversion to start working in the same way, following a process.

The analysis also included the degree of maturity of the communication between departments in the IT-CC. As such, it became clear that there was an inadequate communication between the marketing department and the development department, visible since not all developers were in direct contact with the customers, but with some mediators communicating with all the people. Since the contact between the customer and developers do not always happen in the first hand, information can be filtered. It is important that the IT-CC understands this in order to change and embrace this new approach. If the IT-CC loses these details, they will continue to miss requirements and struggle with how they should satisfy the customer. Changing the approach will help them in the right direction.

Last but not least, the performed analysis indicated that the IT-CC used only partially defined methodology, mainly pushed by individuals, but not as a well-defined strategy of the IT-CC. At this stage, knowledge and experience of the managing of methodology was generally limited, without focused pre-selection of methods for development, placing IT-CC at level two. The IT-CC is in use of methods, which is relevant but it is still up to the developer to decide which to use. This may lead, however, to situations where the developer may not use the most suitable and efficient methods at one time, which can generate unnecessary or superfluous work.

Although some established agile software development processes are described in the literature review and were considered as suitable for the purpose of the project, was later on considered not to be used after the results of the analysis. The reason behind our decision was that the IT-CC's current approach in projects and construction of the organization, made them not suitable to face the changes these processes would require. The IT-CC's projects are usually very small. We believe that this type of methodology and processes would have been useful in bigger projects. While many of their projects are only "solo projects", these approaches are hard to implement when they focus on team development. When talking with the organization, their members expressed a wish of having a process and working routine, a matter that would be easy to implement and that they could use, even with their tight time budget. Therefore, we decided to implement a process that was customized for the company, still using elements of the agile approaches, such as close customer contact, early prototyping and iterative work. We rather implement a process they decide to address, than a best practice approach, they would not be able to attach.

Considering the above, we decided to design a new user-centered SDP that would meet customers' needs, focusing on reaching greater customer satisfaction. The process was agreed to be tested in a pilot study at one of the customers of the IT-CC. The advantages calling for this testing was that the communication with the customer was closer and the developed process tested to see which parts worked properly and which did not. Moreover, it was expected that the results obtained through the pilot study would give benefits for the IT-CC. Since we had the opportunity to perform a pilot study, we had the chance to see how our process worked in practice and not just in theory. Without having had this opportunity, we would never been able to justify why they should use it. Not having a pilot study would have increased the risk that the process will not be used, when we deliver it to them. The pilot study resulted in a customized process, as we could see what was appropriate or not.

The pilot study was conducted as a live project at one of the IT-CC's customers, an electrical wholesale enterprise. The project was conducted in communication with the development team and the project manager from the IT-CC, with us working as usability consultants. The work included customer visits, user-interface design and guiding the developers from the IT-CC towards a result with high user focus. The methods and guidelines used were selected according to the developed user-centered process we aimed to test. The result was very different than what we had thought beforehand. The impact of time and finances were major factors that made us compress the process we created. Some methods did not give enough value to the project. One thing we had as evidence was the customer's positive feedback of being a part of the development. The customer had a positive impression from being a relevant input provider from the beginning.

The pilot study began with an initiating meeting. We carried out interviews with users, together with an open meeting about more technical details. Already in this stage some few paper prototypes were developed. The paper prototypes were shown and discussed with the users. Back at the office, an internal introductory meeting was conducted to start the project within the organization, with responsibilities being divided between the project members and they received the information from the meeting with the customer. Because of that the marketing department early could understand what kind of problem the customer had, the information to developers could be expressed at an earlier stage. By having a developer participating in the first meeting the developer could take advantage of the initial information. This allows for earlier prototypes in the development. This is a benefit when it comes to be efficient and take advantage of the time the company has together with customers.

A second meeting with the customer's users was held as soon as a prototype draft was complete. It was held as a focus group, discussing the graphical interface and the additional functions in the prototype. At the end of the meeting, observations were done with each of the end-users. The result from this usability test was summarized and presented to the clients for sale and further implementation. This was a very important part in the pilot study where we had a lot of feedback from customers, which the IT-CC did not usually get, because they did not use these types of evaluating meetings. It also resulted in that this meeting opened for the opportunity to present technical possibilities, the customer was not aware of that they existed. In this way, the customer has the ability to assess if there is anything further they would like to implement and if not the IT-CC had shown a large commitment and a large interest for their customers.

When testing the developed user-centered SDP in the pilot study, conclusions about the first version's relevance came to our attention. One advantage was that prototypes could be presented for users at an earlier stage in the pilot study, because the customer was aware of their problem. The reason why we did not implement earlier prototyping in the first version of the process could depend on that we are used to use standards and methodology for creating prototypes and perform actions in development. This may have been the reason why we did not foresee it in the first version. We saw the benefits of presenting ideas early, and the user can see what is technically possible and influence along this.

In the pilot study, the proposal was to create an application with some specified functions and requirements in forty hours. Implementing the functions and features that were found during the interviews and observations with the users were not taken into account, instead they were

offered as additional sale. The customer denied some functions and chose other to implement. Without these tests we had never been able to offer additional sale to the customer, therefore they are very good to have in the process. It will help the IT-CC in future projects. But on the other hand if the IT-CC had instead provided and sold in a user-focused working routine already in the first sales stage, it might have resulted in better sale from the beginning. It is better if they can present the IT-CC's new approach in the sales stage and offer the approach and customer focus from the beginning. This would instead create more user-friendly products and it is the customer's choice how they want it. Instead the IT-CC shows from the beginning that they are customer-oriented and focused.

6.1 Problems encountered

During the performance of the master thesis we have faced different issues. We have strived to solve these issues in order to maintain the quality.

One of the problems we encountered in the pre-study was that we could not establish contact and find cooperation with more than one of the IT-CC's customers. The customers were not willing to give time for us. This might have resulted in a subjective view from only one of the client's. However, we also gained insight in the organization itself, where we could distinguish their working routines and opinions. This may have given us a more correct picture of the IT-CC.

When we entered the pilot study as usability consultants, the sale was already approved by the customer. The problem was that the new usability approach was not pitched to the customer. This resulted that the functions and features that were found during the tests could not be implemented immediately and instead were forced to be offered as additional sales. This was not our initiating idea using a usability approach. It also resulted that we could not implement all parts of the developed process due to the time budget of the customer.

If we as usability consultants had been able to influence the IT-CC's organization, we had kept more methods for usability. Since the time budget was short, we had to shorten the process within their restrictions. However, this resulted in a customized process that might make it easier for the IT-CC to implement it to their organization.

6.2 Further advice

After finishing the thesis, the work can still continue at the IT-CC, which also is our advice. The first step is to implement the entire process of the company. This includes educating the developers in the new approach and educates the project managers to control and manage the process. The IT-CC's approach has to change, and it may take a long time before the IT-CC is ready to move on to the next step. The best way continue improving constantly, is to designate a process owner, responsible for the process compliance and improvement. The project owner has the possibility to pilot more projects, in which they can customize the process further.

The next step is a decision making whether or not they will work with the improvement strategy. The IT-CC has all the tools they need to continue the work towards better working routines. If they decide to work following the strategy, they will continue to improve, both in terms of process, methods and staffing.

Further advice could also include that the IT-CC actively improves the communication with the customer. This applies to all departments in the company in order to raise the common understanding in the organization. In this way the understanding of the user will be maintained.

7. Conclusions

Since we introduced different prototype solutions, a routine the IT-CC normally did not perform, we made the customer aware of what was technically possible to create. By informing the customer of the possibilities, it may generate additional sales.

The possibility to evaluate the user-centered software development process through a pilot study was extremely valuable. Justify its impact would have been difficult without it. The evaluation resulted in a customized process.

The pilot study was the influencing factor that made the IT-CC understand the importance of working in a user-centered perspective.

In the pilot study, we experienced how much time and budget affected the process we performed, making us realize that everything has its restrictions.

Provided implementation of consciousness about usability resulted in a greater understanding of the IT-CC customer's needs, in higher profit and in how they should plan, design, and improve their developing process further.

8. References

- Abras, C., Maloney-Krichmar, D., Preece, J., (2004), User-Centered Design, In Bainbridge, W. Encyclopedia of Human-Computer Interaction, Thousand Oaks: Sage Publications, Available at: <http://www.e-learning.co.il/home/pdf/4.pdf> [2010-10-13]
- Andersson L., Nordgren F., (2010), Riktlinjer för behovs- och kravhantering, Fallstudie hos Totalförsvarets forskningsinstitut, Available at: <http://www.iei.liu.se/is/edu/courses/725a22/seminarier-2010/1.201835/10M15AnderssonNordgrenManus2.pdf> [2010-10-11]
- Bernard, Russell H., (2002), Research Methods in Anthropology: Qualitative and Quantitative Methods, Walnut Creek: AltaMira Press
- Graneheim U.H., Lundman B., (2003), Qualitative content analysis in nursing research: concepts, procedures and measures to achieve trustworthiness Department of Nursing, Umeå University, Umeå 90187, Sweden, Available at: http://mindfull.spc.org/vaughan/talks/ns_kat/TrustworthinessQualitativeMethodsReview.pdf [2010-10-13]
- Gulliksen, J. & Göransson, B., (2002), Användarcentrerad systemdesign, en process med fokus på användare och användbarhet, Lund: Studentlitteratur.
- Gulliksen J., Göransson B. et al. (2003), Key principles for User-centered system design. Dept. for HCI/IT, Uppsala Universitet, Available at: <http://www.it.uu.se/research/hci/acsd/KeyPrinciplesForUCSD.pdf> [2010-09-27]
- Kaulio M., Karlsson M., Grubb H., Mellby C., (1999), PRE, Product Requirements Engineering, Kundförståelse i produktutvecklingen, IVF, Mölndal
- Nielsen J., (1993), Usability Engineering, Focus Groups, Published by Academic Press, Inc, San Diego
- Norman D., (1986), Cognitive Engineering, User-centered System Design, Lawrence Erlbaum Associates Inc., Hillsdale, New Jersey.
- Osman A., Baharin H., Ismail M., Jusoff K. (2009) Paper Prototyping as a Rapid Participatory Design Technique, Available at: <http://ccsenet.org/journal/index.php/cis/article/viewFile/3418/3096> [2010-10-13]
- Perez G.A., (2003), Model consistency in the object oriented software development process, Perceived in OOPSLA'03: Companion of the 18th annual ACM SIGPLAN conference, p.398-399, October 26–30, 2003, Anaheim, California, USA. ACM 1-58113-751-6/03/0010
- Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. & Carey, T., (1994), Human-Computer Interaction, Addison Wesley Publishing Company, Workingham, England

- Rational the software development company (1998), Rational Unified Process, Best Practices for Software Development Teams, Available at: www.ibm.com/.../rational/library/.../1251_bestpractices_TP026B.pdf [2010-10-06]
- Sharp H., et al., (2007), Interaction design- beyond human computer interaction, 2nd edition, John Wiley & Sons Inc.
- Stapleton J., (1997) DSDM, dynamic systems development method: the method in practice, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA
- Tec-Ed (2009), User-centered Design. Available at: http://www.teced.com/services_design_user_detail.html (last accessed 2010-09-27)
- Van Someren M., Barnard Y., Sandberg J., (1994), THE THINK ALOUD METHOD - A practical guide to modelling cognitive processes, Department of Social Science Informatics, University of Amsterdam, Published by Academic Press, London, 1994, ISBN 0-12-714270-3

Methods and Guidelines

Sales pitches

In order to give your clients better understanding why your projects are carried out with focus on usability, you can use the following arguments. Note that they do not give a guarantee for better or additional sales!

"Our products will not only satisfy your expressed needs, but also your unspoken needs."

"The reason why we are working with a careful preparatory work is that it reduces the occurrence of unintended consequences at the end of development. We are right from the start, simple as that!"

"It is important for us at the IT-CC to understand your needs, to be able to satisfy you in the correct way."

"Developing an understanding early in our project, will not only improve the system but also reduce delays and rework in our specifications."

"Understanding early in the process what you need, will most certain reduce the variable costs."

"You are all users to the system, where you all will benefit from it in the end. Therefore, we see the importance that you have a central stage in the development"

"Focus on usability provides a system that is specifically tailored to your users. This makes the practical part easy to learn and less time consuming."

Interview template

The interview template includes questions that are relevant to mention during an interview. Not all questions need to be answered, but a selection of the best suited for the situation should be reconsidered. Additional questions can be included if they are found important.

General questions - in order to get a glimpse of how the system works at present time

- What is the purpose of the system today?
- How often do you use the system? (on a daily basis, weekly)
- How much time do users spend when using the system today?
- What steps are needed to perform tasks in the system?
- In what environment is the system used?
- Do users work together with each other when completing tasks in the system?
- What resources, computer systems, products, form, etc., does the user need to perform or complete tasks in the system?
- Are there any critical situations that may aggravate the interaction with the system?
- How can the situation and the informal support be improved?

User-related issues - To gain an insight into whom the user actually is.

- What different users are available?
- How many users are interacting with the system?
- Do users have different needs or desires?
- What level of experience does the user have that are going to interact with the system?
- What education and computer experience does the user have?

Tip

- Adapt the questions you intend to put to the customer.
- Do not forget to focus on the end user.

Paper prototyping

Paper prototyping is the first method you will be using when trying to visualize the system that is being evolved. It can help you as a developer to be able to present several different interface options in a very short time.

Benefits

- Quick and easy technique, in which adjustments can be performed very quickly.
- Customers have the opportunity to influence the appearance and navigation of the prototype. As they know that is not yet fully developed, they can express their own opinions.
- The customer gets involved early in the project and feel more important.

Performance

Paper prototype is based on sketches with pen and paper. The sketches may well be a scenario-based so that you have the ability to navigate through the prototype or perform a task. Have a great canvas that gives you a good overview of many different parts of the prototype display. Begin to sketch a basic skeleton, where the prototype can always be assumed. The prototype shall be developed with the requirements and functionality in mind, but do not be afraid to draw the interface looks. Sketch several options. Based on the basic skeleton, add post-its and the new leaves that represent other parts of the system. Imagine yourself as a user and get a sense of the navigation and structure actually feels natural. When a number of sketches and proposals are developed, it is time for prototype testing.

Tip

- Do not spend too much time to make it look good. Make only rough sketches

Material

- Paper
- Pens
- Post-its

Semi-structured interview

The "Semi-Structured Interview" is a data collection method that is the foundation to get a better understanding of your customer. It will help you as developers to get a better understanding of who the customer is, his circumstances and his needs. "Semi-structured interviews" are based on a number of carefully selected issues to be addressed. Beyond these issues, there is also room for further questions that arise during the interview.

Benefits

- Significant questions will be answered.
- There is room for additional questions.
- Customer relationship is strengthened.

Performance

Before you perform the interview, make sure that you know who the customer is and what he works with. This will prepare you mentally, so you do not come to the interview unprepared. In that way you will be prepared on how to ask your questions and how far into the interview, you have come.

Record and write down what is said during the interview. If you have a colleague with, allocating work areas is to prefer. One can ask the questions and adjust the tempo of the interview, while the other one document what is said. If you are on your own, be calm and ask for breaks if you do not have time to document everything that is said. If it is short on time, bring a recording tool, so you have the opportunity to document afterwards.

Tip

- Use the interview template, M&G #2, where good basic questions are found.
- Avoid leading questions, which may be misinterpreted or not entirely accurate from the real situation.
- If you do not have time to document everything that is said you can tell your client: "Now you said so much good stuff so I cannot keep up! Wait a bit, so I have time to write down everything."
- Do not bring with you more questions than the time frame allows.

Material

- Laptop
- Paper and pen
- Interview questions

Content analysis

"Content analysis" is a method you can use to compile large amounts of data from different data collection methods. Data can come from interviews, questionnaires, quotations, recorded or filmed media. Using the method as a developer, you can analyze data and clarify it.

Benefits

- Can compile large amounts of data retrieved from different collection methods.
- Trends and similarities are detected and categorized easily.

Performance

Create a document in which you write, in paragraph form, all the sentences and responses from the interviews exactly as they have been written down before. If you have recorded data dictate everything that has been said. The part you are now creating is called "meaning units". Read all sentences thoroughly.

Based on the sentences, make an interpretation of the underlying meaning of them. Write down the interpretations in a new column, this column is called "interpreted meaning unit". When you have written down all the interpretations, it is easier to see the similarities and differences.

The interpreted meaning units will now be divided into subthemes, which are named according to the common denominators they consist of. Create a new column with the subthemes and arrange them in a grouped order.

Example:

Meaning units	Interpreted meaning unit	Subtema
I think it's unclear with the date. One must count themselves until what date it is.	The interface is not quite constant, which means that the user has to think for himself in order to locate the correct date.	Indistinct interface.

Tip

- Perform content analysis, preferably with a colleague and talk through the "breakdowns" that you do. This gives you a greater perspective and even more interpretations of the meaning units.

Personas

The method "Personas" is a tool for you to get a clear picture of what different types users exist in the system. A persona is a fictional person representing the actual user to an IT system, a specific user type.

Benefits

- You get an insight into which the real end users are.
- You become aware that users are made up of different individuals and have different needs, habits and preferences.
- It becomes easy for you to pass on to the different users and your colleagues what aim the system has.
- It becomes easy for you to convey your understanding of the different target groups to the customer.

Performance

To be able to construct "Personas", it requires that you build up knowledge about who the real users are. In addition, you must have knowledge of the user's habits and experience, combined with a decision about which are the most important target groups. The decision may come from the customer or obtained through knowledge of who is the most frequent user.

You obtain knowledge by having interviews with users and examining how their approach works today. It is important that you take part of several user situations. If this is not possible, ask someone else who has a deep understanding of the users, to inform you about it.

In the next step you create fictional people who have the same needs and requirements as the real target groups, in order to create stereotypes.

An example of a persona is Peter, who has developed a web-based time-report system.

"Peter is 37 and has been working as an consultant at company X for the last 3 years. He was working as a Java developer the first two years but has in recent years worked as a project manager in several different projects. He works in several different projects simultaneously and travels a lot. His goal is to time report his hours in the various projects in an efficient manner and also to report his travel time to and from his customers and give the department the right information about his working hours."

Tip

- Base your "Persona" in the knowledge you have about the actual users of the system, don't make up or have preconceived ideas!
- Try to be clear and instructive when you create your personas so that others around you also can take advantage of them.

- Use and make use of "Personas" you created in the whole development of the system so they won't be underestimated in development.

User stories

The method "User Stories" are short paragraphs that contain descriptive text about what a system will perform. They are used to in a clear and simple way, describe the overall requirements that a system has.

Benefits

- "User Stories" express the client's needs.
- "User Stories" are brief and contain no technical details.
- "User Stories" provide a common understanding for the customers and developers.

Performance

In order to perform "User Stories", the customer shall have an active role together with you as a developer. This can be done in correlation with the external project meeting. It is important that the customer puts the user stories according to their wishes. The focus is on what is needed in the system, not how it should be resolved by the developers. This contributes to the short and simple descriptions of what the system should perform. User stories can be added as new demands arise.

To write a user-history you can use the following format:

As ... [Actor] ... I want to ... [Targets] ... so that ... [Reasons].

By following this template, you get developers who not only know what the system will perform, but also why the user thinks it is an important function. Here are a few examples of "User Stories":

- "As a user I want to register my personal information so that it is stored for future use."
- "As a user, I would be able to change my personal information so that I do not need to register a new user if I need to change the address."

Tip

- "User Stories" should be concise and be represented as a basis for all requirements.
- "User Stories" should be written so that those requirements be interpreted in the same way by all who reads them.

Requirements specification

A requirements specification consists of functional and non-functional requirements. The functional requirements describe the functionality of what the system should contain, while the non-functional requirements include the conditions required for the functional requirements to be met.

Benefits

- The requirements are documented and you can always refer back to them to make necessary updates.
- A clear specification gives you control over the system's complexity and its time and resource use.
- Gives a common understanding to the customers and developers.
- A clear specification facilitates the testing because you have documented what the system should do.

Performance

The basis of a detailed requirements specification from the customer is that they are aware of what you develop and that you are aware of what is supposed to be developed. It is important that all stakeholders are involved in designing the specifications when requirements differ in importance for different stakeholders. Based on the specification an estimation plan can be created to visualize the complexity of the project.

To begin a requirements specification, reconciliation with the project team and customers must be done. Precede the user stories that already exist to be able to retrieve the functional requirements for the system. The requirements will be documented in a list to be amended and updated as the project moves on. The functional requirements will include the requirements that describe what the system should do. The non-functional requirements will include the requirements that describe how the system should work in, performance, interface, usability, security and availability.

Tip

- The requirements should be written so that all stakeholders understand the meaning.
- Technical jargon should be avoided as far as possible.
- Highlight for the client how important it is to meet up to discuss the requirements management.

Target group analysis

A target group analysis involves collecting data on all types of target groups with help of, for instance, interviews, see page. 4. After collecting, analyze the data that is received by setting up the similarities and differences between the different target groups.

Benefits

- You must understand the types of target groups that a system has and also what demands they have.
- You can see the similarities and differences between the groups.
- It simplifies the creation of feeds for the different target groups.

Performance

In the implementation of a target group analysis, you assume from the "personas" you created earlier. In the various "personas" that you created you can now visualize their different preferences. Write down all the requirements and habits of the different target groups and have them rated colored, one color per target group. Now, carry out a content analysis, see page 5, in order to compile subthemes. That way you can see the themes, demands or requests by the various target groups and which parts they do not share, see the example below.

Meaning unit	Interpreted meaning unit	Sub-theme
I normally search for what I'm looking for. I would really like to have a search bar.	A search bar is demanded.	Search bar
A search bar should exist	A search bar is demanded.	
You should be able to edit or remove personal information.	Would like to edit or remove personal information.	Edit or remove personal information.
You should be able to register yourself as a user.	Login is demanded.	Login
There has to be a login.	Login is demanded.	
There has to be a login,	Login is demanded.	
I would like to pay with my credit card. I find paypal and invoice annoying.	Target group demands payment with credit card.	Payment method: Credit card
It would be neat if you could pay with invoice. I don't trust payment with credit card over the internet.	Wants to pay with invoice.	Payment method: Invoice Payment

Tip

- Remember to take into account the target groups that are most important to the customer.

Prioritizing requirements

Prioritizing requirements is important to you as a developer. It helps you understand what your customers think is important in the development of the system. By being aware of what is considered important, it is easier for you to address the requirements of the specifications in the right prioritized order.

Benefits

- You understand your customer better and fulfill their demands in the right order.
- You carry out the requirements that have both low cost and high value first.
- You get an insight into how much the customer is willing to pay.

Performance

There are several different methods for prioritizing requirements. Select the method that feels most appropriate for your project. The requirements do often need to be compared both by the customer and yourself as very often, you do not share the same perspective on the project. Let the customer prioritize the requirements according to his needs, then you can do the same kind of priority from your perspective. After this stage, compare the priorities with each other and contact afterward the customer by telephone or meeting to discuss how you two look at the requirements.

"Weightning"

Weighting means that each requirement is given a weight from a given scale. The scale should not be too long, also not too short. 5-8 steps on the scale tend to be suitable to get a good spread between the requirements. After selecting the type of scale, you compile a list in a survey that is linked to every single requirement. Now you let the customer prioritize the requirements with help of the scale.

"The hundred dollar bill"

The hundred dollar bill means that you let the client allocate 100 dollars on a list of demands. This method often gives a greater spread between the requirements in comparison with the "weighting" method. This is a quick and easy method and can be monitored easily by phone or e-mail. This method is suitable when the requirement list is long (10-15 requirements), especially if you need to compare all of them with each other.

"Priority against two criterias"

In this method, bring up all the requirements and list them in an Excel spreadsheet. Allow the customer to take account of their user value, and then you get developers to take account of the development cost. The scale goes from 1-4, where 1 is the least important one to the user but is implemented at low cost and there 4 is the most important, but implemented at a high cost. The focus here is on the requirements of high value to users with a low development cost. The other requirements can be expected to develop unless the customer has a different opinion.

Example:

ID	Header	User value	Development cost
1	Login	2	3
2	Search bar	4	2
3	The possibility to save work	3	3

Tip

- Remember to take account of your client's desire in the first place.
- Discuss and give your comments to the client, especially because your perspectives may differ.

Concept and prototype testing

Concept and prototype testing is a general term and includes various types of testing. As a developer you can use it for user testing interfaces and be able to validate and confirm that you are on the right track. You can also obtain feedback on quite ready prototypes of the system that you are developing.

Benefits

- Allows the customer to be involved from the start. Creating a relationship with customers in the very beginning will increase their confidence.
- You get feedback from real users early in the development.

Performance

The tests can be carried out in the customer's real environment, but can also be performed at the IT-CC. It's good if you can implement tests to multiple users from different target groups.

Allow customers to review and test the prototypes you have developed, this can be done already at the stage where you produce paper prototypes. Prepare a questionnaire that you can give out to the people who interacted with the prototype.

If there is sufficient time, you can e-mail instead the prototype and its questionnaire. Let the customer give feedback if there is something they had in mind that is not included or mentioned in the questionnaire.

Tip

- The choice of test environment can be significant. It often returns the best feedback when it is tested in its natural environment.
- Do user tests with people that you have met before as your relationship already has grown.
- This type of test can also be done internally at the IT-CC to receive more valuable feedback.

Observation

"Observation" is a data collection method based on observing and documenting how a user interacts with a system. An observation can be done in several ways, but the most traditional is that the observer sits in the background and documents each step of the users' interaction with the system.

Benefits

- As a developer, you get a better understanding how the user navigates in the system.
- All interaction is observed in real time, reality emerges.
- Details that would normally be discarded is now documented as it happens in this very moment. This also includes details that are reformed in time, when our memory fails us.

Performance

Before observation, create a number of scenarios and tasks that the user will be able to perform. This information should not be of complex nature or misleading, they should be handled from a natural task.

The observation should be done in a silent environment with as little disturbance as possible. This is because the user should not become distracted of anything and only focus on the system. Feel free to bring a colleague whose principal task will be to document. An observation can also be recorded to include details such as how the user moves the mouse on the screen, the user's facial expression, etc.. When the user performs a task, let him adjust the pace of the interaction. Never give any clues how to use solve a problem or explain why things are structured in a certain way. When the observation is complete let the user tell you if there's anything special he / she wants to add, both negative and positive feedback.

Tips

- Make sure you have plenty of time.
- Ensure that the tasks are realistic and achievable.
- Let the user solve the tasks himself; never give advice or personal views.

Think aloud

"Think aloud" is a method used to gather data when usability testing. It can be used in combination with other methods such as observation (p. 13) and the concept and prototype testing (p. 12).

Benefits

- Possibility to receive information that might not arise with the other methods such as feelings and thoughts.
- The participants express their current thoughts, which provide more detailed feedback.

Performance

"Think aloud" forces the person to think aloud throughout the whole test. The purpose of the "think aloud" is to allow the user to state what he feels in the exact moment while interacting with the system. While the test is progressing you should encourage the tester to tell you what he's seeing, thinking, doing and feeling.

Tip

- Do not state leading questions.
- Document precisely. Do not add or remove information that the user hasn't expressed.
- Feel free to record
- Bring a colleague.

Focus groups

"Focus groups" is a method performed by a number of users of a system. The idea of focus groups is that everyone's voices should be heard. That way you get the distinct and common views of what they think.

Benefits

- Space for discussion is on hand which can give a higher understanding of think of the system.
- Opportunity is provided for users to "educate" each other on how they normally interact themselves.
- Time-effective way to gather information from a lot of users at the same time.

Performance

Make sure to plan a meeting in advance with a number of end users of the system. Try to meet different types of end users, users that have different needs and levels of experience. Make sure you have prepared different themes and situations that may occur in the system so there is always a predefined topic to talk about. Allow users to speak themselves and keep it alive and do interfere only if you notice that the discussion is losing focus. Don't express your own opinions, let the users work on that part themselves.

Tip

- Make sure that everyone talks.
- Bring a colleague.

Cognitive walkthrough

"Cognitive walkthrough" is a method used to identify the interaction problems that may arise from the use of a system. It is based on a task analysis that identifies in detail the stages of the task that has to be performed to be able to complete a task.

Benefits

- Weaknesses of the system are easily found as every step that is taken is considered.
- Unnatural interactions are identified.

Performance

Create different tasks that the user must perform. Each task contains several steps which are divided separately. One step is described by a user action (UA), an action made by the user, which in turn generates a system response, an action that the system does (SR). At each step performed four predefined answers must be answered with yes or no:

- A. Are the results of the user action correct?
- B. Is it visible for the user if the action or operation is available?
- C. When the user has found what seems to be the right operation to continue its task, is it completely obvious that it is the correct action?
- D. Will the user understand the feedback that system is responding after processing an action?

If you answer 'Yes' to a question, continue to the next one or the next step in the task. If you answer 'No' to a question, describe why you it is so. In order to ascertain how it works practically, you can see an example below. The example is based on a grocery online store:

Task 1: Add 1 kg Entrecote into the shopping cart.

UA1: Click on 'Meat'

SR1: Viewing a page with the category of 'Meat'

Answers to questions A: Yes B: Yes C: No D: Yes

Description of the answer 'No' in C: Because there is only one sub-category of 'Meat' it can be seen as an unnecessary "extra step" of having to click on the 'meat' (as in category) again. The reason why the program looks in this way is for the purpose of expanding, but at present time it is an unnecessary step giving risk that users become confused and irritated.

Tip

Bring a colleague

Knowledge bank

A 'knowledge bank' is a good tool to internally share the experiences you received during a project. It may contain features of code that can be useful in other projects or problems you encountered and how you solved them. The purpose of the knowledge bank is that you should not have to make the same mistake again, in other words not needing to reinvent the 'wheel' again.

Benefits

- It is time efficient to reuse code and take gain knowledge of your colleagues.
- The knowledge is shared for all future, even if a colleague works somewhere else.
- You do not make each other's mistakes, as you might be aware of them by now.

Performance

Use SharePoint as a portal to share your knowledge with your colleagues.

Tip

- Use a standard when writing to the portal, so everyone easily finds what they are looking for.
- Do not wait to fill in relevant information to the knowledge bank. Try to document as soon as possible so nothing is forgotten.

Appendix B – PRE - Improvement strategy

PRE – Improvement strategy

Products requirement engineering

Products requirement engineering, PRE, är ett verktyg som används som underlag till hur utvecklingsprocesser ska hanteras. PRE fungerar som handledning för företag som vill förbättra sin kundhantering i samband med produktutveckling. PRE fokuserar på tre områden, processen, bemanning och metodik. Utifrån dessa tre områden utförs en s.k. självdiagnos som visar vilken mognadsgrad företaget befinner sig i nuläget. Utifrån självdiagnosen skapas en förbättringsstrategi för företaget, som vägleder hur man ska arbeta för att öka mognadsgraden.

Hantering av kundkrav

Vid förbättring av kundkravshantering fokuserar man på tre olika områden. Områdena behandlar processen för kundkrav, hur man bemannar projekt och vilka metoder som effektiviserar och förtydligar arbetet med kundkrav.

Processen för att hantera kundkrav

För att en process ska kännetecknas som bra vid hantering av kundkrav så måste ett antal kriterier följas. Kriterierna syftar till att skapa en bra kontakt med kund och ett tidigt fokus dess krav.

Processen för att hantera kundkrav ska ha en tydlig kund med ett tydligt mål. Med andra ord fokusera på slutanvändarens behov och önskemål. På detta sätt skapas mervärde för processens kund. Processen ska vara standardiserad och visualiserad inom företaget. Det måste även finnas en tydlig start och ett tydligt slut, där det är beskrivet hur alla aktiviteter ska hanteras vid projekt. För att processen ständigt ska kunna förbättras är det viktigt att det finns en utsedd ansvarig som kontinuerligt följer upp, vilket möjliggör implementering av förbättringar. En utsedd ansvarig ska också definiera ett antal mätetal som anses viktiga för utvecklingsprocessen. Mätningar följs upp av den ansvarige.

Målet med en definierad kundkravshanteringsprocess är att bygga en förståelse för kunden och vilka önskemål och behov den har. För att uppnå kundförståelse finns det ett antal perspektiv att ta till vara på. För det första är det viktigt att man definierar målgruppen för den produkt man ska ta fram. På så sätt kan man få en uppfattning om vem man behöver träffa och vilken typ av information man behöver samla in.

I nästa steg ska en analys av information samlats in och genom analys generera tekniska lösningar till hur man ska lösa användarens problem. En formulering och en prioritering av kundkraven utförs. För att processen ska bli fullständig ska en verifiering av resultatet genomföras. Det kan göras under flera steg i processen för att man inte ska komma för långt ifrån användarnas verklighet.

Bemanning för att hantera kundkrav

För att hantera kundkrav krävs en genomtänkt och bra bemanning av projekt. Bra bemanning kännetecknas av samverkan inom organisationen och en god förmåga att driva projektet framåt. Detta kräver att såväl marknads- samt utvecklingsavdelning har aktiv kundkontakt.

Till att börja med måste organisationen ha förmågan att driva projekt. Det ska finnas en tydlig struktur i projektet där man vet start respektive slut. Organisationen ska vara medveten om projektets intressenter, slutkunden, beställaren respektive projektgruppen själv. För en bra bemanning ska kommunikationen inom företaget vara tvärfunktionell. Här är det viktigt att marknads- och utvecklingsavdelningen samverkar i ett tidigt stadie. Företaget måste se skillnaden mellan att arbeta i team och arbeta parallellt i olika aktiviteter. Istället för att arbeta var och en för sig och sedan stämma av, så arbetar teamet tillsammans på en och samma uppgift för att kontinuerligt erhålla statusuppdateringar. Organisationens ska också tillsammans skapa en kollektiv förståelse. Med detta menas att projektledare och projektmedlemmar arbetar mot samma mål på ett effektivt sätt där intervjuer sker parvis och analysarbetet aldrig görs av en enskild individ.

För att hantera kundkrav är det av vikt att alla inom projektet har aktiv kundkontakt för att driva den kollektiva förståelsen inom organisationen. Information om kundkrav inhämtas alltid först hos kund där en projektmedlem med teknisk kunnsighet ska medverka för att kunna avgöra huruvida de tekniska vägvalen ska hanteras. Förstahandsinformationen får aldrig påverkas av tolkningar, vilket kan leda till att den tappar sin innerbörd.

Inom projektgruppen ska medlemmarna arbeta det mesta av sin tid med projektet. För att få ut den bästa kompetens och effektivitet är det, i större projektgrupper, viktigt att medvetet välja medlemmar efter människors olika erfarenheter, egenskaper och kompetenser.

Metoder för att hantera kundkrav

Det som karakteriserar en bra metodanvändning för att hantera kundkrav handlar om att inte bara välja den metod som genomför aktiviteten bäst utan även om hur mycket tid och resurser som finns tillgängligt och krävs. Specifika metoder passar olika företag och olika projekt.

För att bli riktigt bra på kundkravshantering krävs det att företaget börjar att använda metoder och införa de i sitt arbetssätt. Det kräver enkla metoder och att man tar enkla steg mot en bättre kundkravshantering. När man väl börjat inkludera metoder i sin process kan företaget kontinuerligt förbättra de metoder man använder och bygga på med fler och mer komplicerade metoder. Det är också viktigt att fånga de krav kunder har med hjälp av rätt metoder. Det är viktigt att inte bara se till generella metoder som intervjuer, utan också använda andra som observationer för att se vanor och beteenden hos användarna. Det är då man får fram nya krav, så kallad överraskningskrav som är de krav som kunden inte själv är medveten om. Det krävs också att man genomför metoder som täcker de delar i systemet som man vill lägga mest fokus på. Dessutom ska man kunna kombinera och använda sig av metoder som täcker flera steg i processen. Man bör också välja metoder med direktkontakt med användarna och på så sätt är man medveten om att informationen inte är filtrerad.

Självdiagnos

För att mäta vilken nivå Företaget förhåller sig inom kundkravshantering har vi utfört mätningar med hjälp av PREs självdiagnos. Självdiagnos kan användas för att mäta vilken mognad företaget har i dagsläget och även vilka delar som företaget saknar för att uppnå en högre nivå av mognadsgrad.

Självdiagnosen är indelad i tre olika områden, process för kundkravshantering, bemanning i projekt och metoder för kundkravshantering. Alla tre områden förhåller sig till en nivåskala som löper från 1-4, där 1 är lägst och 4 är högst i mognadsgrad. Varje nivå kräver att företaget uppfyller specifika kriterier för att öka i mognad. Om en av kriterierna i ett av områdena inte uppnår en specifik nivå, avstannar den till den lägre.

Företaget - självdiagnos

Som tidigare nämnt har vi utfört en självdiagnos för att avgöra var Företaget placerar sig i mognadsgrad. Utifrån intervjuer med personal och kunder har vi analyserat och tagit fram ett resultat som sammanställts i matrisen nedan.

Företaget	Process	Bemanning	Metoder	
Nivå 4				Ständigt förbättrat
Nivå 3				Implementerat
Nivå 2				Definierat
Nivå 1				Sporadiskt

Matrisen visar att Företaget har uppnått mognadsgrad 1 i processen för kundkravshantering och mognadsgrad 2 i bemanning respektive metoder för kundkravshantering. Resultatet kanske uppfattas som oroväckande, men det är det inte. Det som resultatet påvisar främst är avsaknaden av vissa detaljer som måste läggas till eller finslipas. Detaljerna som Företaget saknar skulle resultera i förbättringar, och ge en positiv effekt i framtida projekt. Utifrån självdiagnosen kan man se till de aspekter som måste beaktas för att uppnå högre mognadsgrad. Det gör man med hjälp av en förbättringsstrategi.

Förbättringsstrategi

Vid utförd självdiagnos redovisas alltid de kriterier som saknas för att uppnå en högre mognadsgrad. Genom att utföra en förbättringsstrategi finns möjligheten att öka företagets mognad. För att uppnå en högre nivå måste alla tre områden vara på samma nivå och därmed skapa en balans. Om det råder obalans, ska de med lägre nivå höjas först.

Företaget - förbättringsstrategi

Utifrån resultatet i självdiagnosen, har en förbättringsstrategi tagits fram för Företaget. Förbättringsstrategin hjälper oss att beakta de delar inom organisationen som behöver utvecklas ytterligare för att förbättra kvalitén i kundkravshantering.

Företagets matris visar att bemanning och metoder har uppnått nivå 2 medan process är på nivå 1. Det första steget är att åstadkomma balans i områdena, vilket innebär att processmognaden måste höjas till nivå 2. När processmognaden väl uppnått nivå 2, kan förbättringsarbetet med att nå nästa nivå inom områdena återupptas. Områdena ska ökas i nivå i en bestämd ordning, p.g.a. att redan uppnådda nivåer hjälper att höja de kvarvarande. I matrisen nedan redovisas i vilken ordning nivåerna ska höjas. Det sker i prioritering 1 till 7 där den första.

Företaget	Process	Bemanning	Metoder	
Nivå 4	6	5	7	Ständigt förbättrat
Nivå 3	4	2	3	Implementerat
Nivå 2	1			Definierat
Nivå 1				Sporadiskt

Företaget – Strategi

För att Företaget ska kunna utveckla och förbättra sin utveckling i projekt måste förbättringsstrategin genomföras. Genom att arbeta aktivt med de kriterier nivåerna kräver, växer mognadsgraden i företaget. I prioriterad ordning redovisas de steg Företaget måste ta för att uppnå ett effektivt arbetssätt.

Strategi för nivå 2

För att uppnå nivå 2 i processmognad måste Företaget definiera sin process för hur ni hanterar de kundkrav som finns. Dessutom måste enstaka projekt följa denna process.

Företaget har delvis uppnått nivå 2, men uppfyller inte alla kriterier för vad den kräver. En projektgrupp är framtagen med fokus på process och en process har börjat tas fram. Dock så är inte processen fullständig vad det gäller hantering av krav och alla i organisationen är inte medvetna om dess innebörd.

Nästa steg för Företaget är att identifiera huvudsteg, beslutspunkter och aktiviteter för den nya processen och förankra dessa i företaget. Viktiga aktiviteter och insamlingsmetoder ska identifieras. Aktiviteterna samlas sedan ihop till huvudsteg i den framtagna processen. Dessutom måste den nya processen förankras i företaget där alla anställda ska vara medvetna om den och ha en kunskap om processen. Det kommer underlätta implementeringen av den i organisationen. Här är också alla anställdas synpunkter viktiga. Det ska också utvecklas en processmanual där organisationen kan ta del av mallar till aktiviteterna.

Företaget	Process	Bemanning	Metoder	
Nivå 4	6	5	7	Ständigt förbättrat
Nivå 3	4	2	3	Implementerat
Nivå 2	✓			Definierat
Nivå 1				Sporadiskt

Strategi för nivå 3

För att lyfta Företaget från nivå 2 upp till nivå 3 inom området bemanning måste marknad och utveckling aktivt jobba tillsammans. Ett viktigt steg är att utbilda alla projektdeltagare i projektarbete och hur man arbetar i processen. Det är också i denna nivå ni som företag kan börja ställa krav att alla ska ha kundkontakt. Dessutom ska interna möten hållas där alla är involverade och arbetar tillsammans.

För att öka från nivå 2 till 3 inom metodhantering måste fler metoder tillämpas i processen. Kännedom om fler metoder och hur man lär sig att sortera sig fram till den mest lämpliga är nu ett krav. För att lära sig sortera bland alla metoder ska det finnas en metodkatalog så att man lätt ska kunna få en överblick över vad som finns till sitt förfogande. När man har fått en överblick, ska metoderna testas, för att kunna utvärdera resultatet och avgöra vilka metoder som ska förbättras. Det viktiga i det här momentet är att lära känna hur du ska hantera de olika metodkategorierna och vara medveten om att de alltid kan anpassas och kombineras beroende på situation. Fördelen med kombinationer eller samspelet mellan insamlingsmetoder är att det möjliggör att skapa en helhetsbild. Helhetsbilden kan handla om olika viktiga områden såsom slutanvändaren, produkten och kraven. Analytiska metoder kommer normalt till användning när man samlat in tillräckligt mycket relevant data. Det viktigaste är att lyckas hitta en struktur så att uppreppande mönster och gemensamma punkter hittas.

Genom att ha byggt en grund med dessa förutsättningar möjliggörs att resterande arbete blir öppet för metodval. Vissa metoder är knutna till att ha nära kundkontakt medan andra ger möjligheten att inte behöva ta upp en ny kontakt. Det man inte ska glömma är att kontinuerligt ha den nära kontakten till kunden och se till att Ni förstår varandra. På så vis minskar och elimineras missförstånd som kan uppkomma.

För att uppnå nivå 3 inom processen krävs det att den är fullt implementerad och att den används i alla produktutvecklingsprojekt. För att den utvecklade processen ska fungera som det hjälpmedel det är avsett att vara, måste alla inom organisationen förstå och använda sig av den. Som hjälp kan man använda sig av projektledningen som tidigt i utvecklingen kan kontrollera att processen följs. För att nå denna nivå kan det också vara lämpligt att utse en processägare. Denna person ansvarar för att kundkravshanteringen granskas och ständigt förbättras.

Företaget	Process	Bemanning	Metoder	
Nivå 4	6	5	7	Ständigt förbättrat
Nivå 3	✓	✓	✓	Implementerat
Nivå 2	✓			Definierat
Nivå 1				Sporadiskt

Strategi för nivå 4

Att öka företagets bemanningsmognad från nivå 3 till 4 kräver ökad förståelse i konflikthantering och kännedom om sina kollegors personligheter. Det finns en hel del lämpliga förbättringsstrategier man kan följa. Genom att utveckla ett system med resursplanering kan man på ett enkelt sätt få översikt vilka kollegor som finns tillgängliga framöver. På så sätt kan man förbereda framtida projekt smidigare. När man har skapat en god resursplanering ska organisationen lära sig att arbeta effektivt med projektprioritering, för att arbeta så effektivt som möjligt. Håll en hög nivå med benchmarking, där man kontinuerligt jämför sig med andra medlemmar utanför projektgruppen.

Projektledare ska ha en kompetens där de känner sina medlemmar så väl att de ska kunna välja och placera rätt roller på rätt plats. Det skapar en kontrollerad grupp som fungerar bra tillsammans och har den kompetens som behövs för att klara de utmaningar som möts.

Ge utrymme för projektrum där det finns plats för utveckling och kreativa tankar. Bortsett från projektets mål ska det finnas möjlighet för både individen och gruppen att utveckla egna kompetensmål utifrån projektet. Försök få alla projektmedlemmar att hålla sig till ett begränsat antal projekt samtidigt, främst för att få medlemmarna att uppleva en stor del av utvecklingen innan de lämnar den.

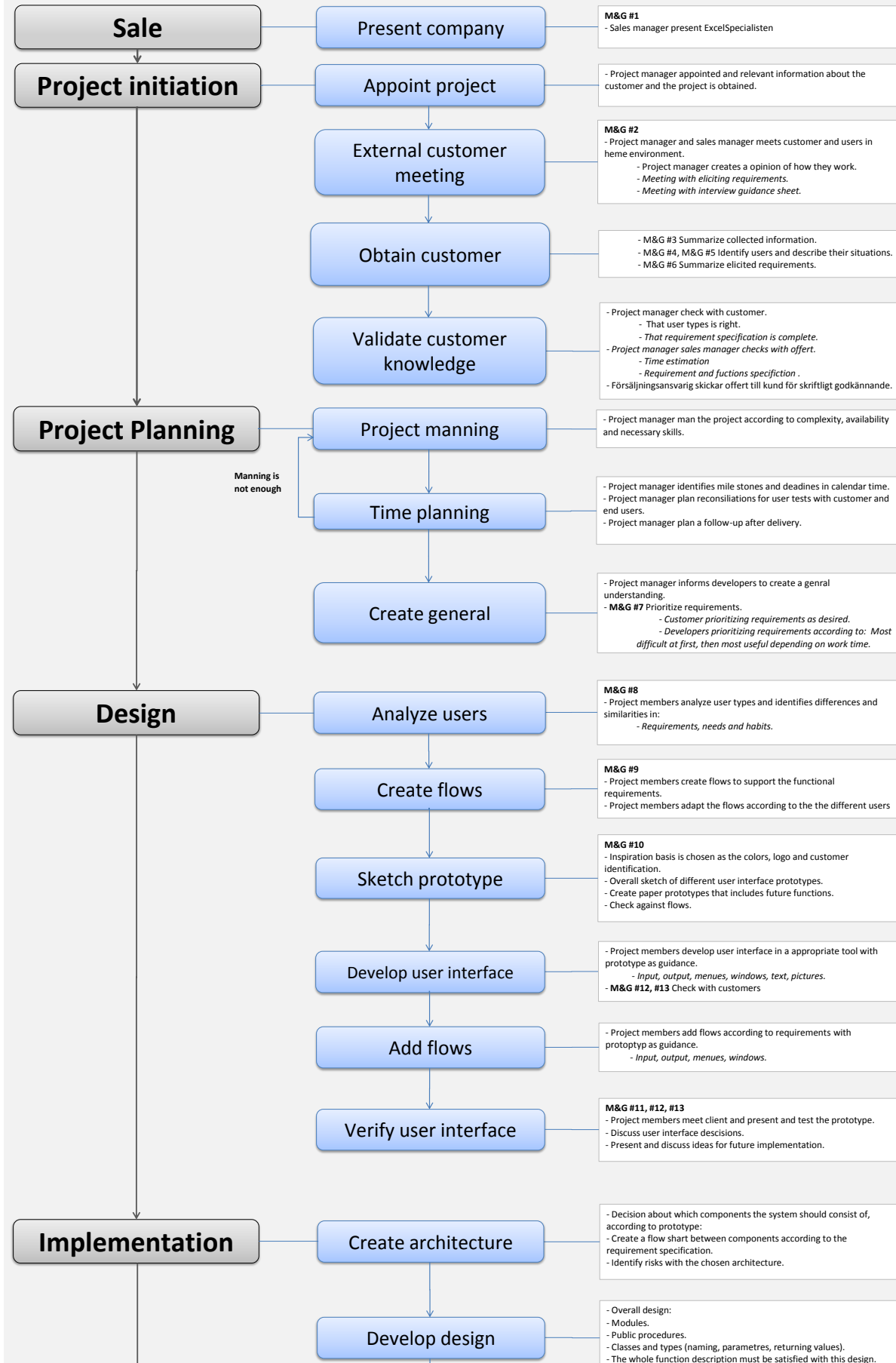
För att uppnå nivå 4 inom området process, krävs det att Företaget ständigt försöker förbättra processen vad det gäller hantering av kundkrav. Det krävs även att Företaget mäter sin processprestanda och jämför hur ni arbetar i jämförelse med andra företag.

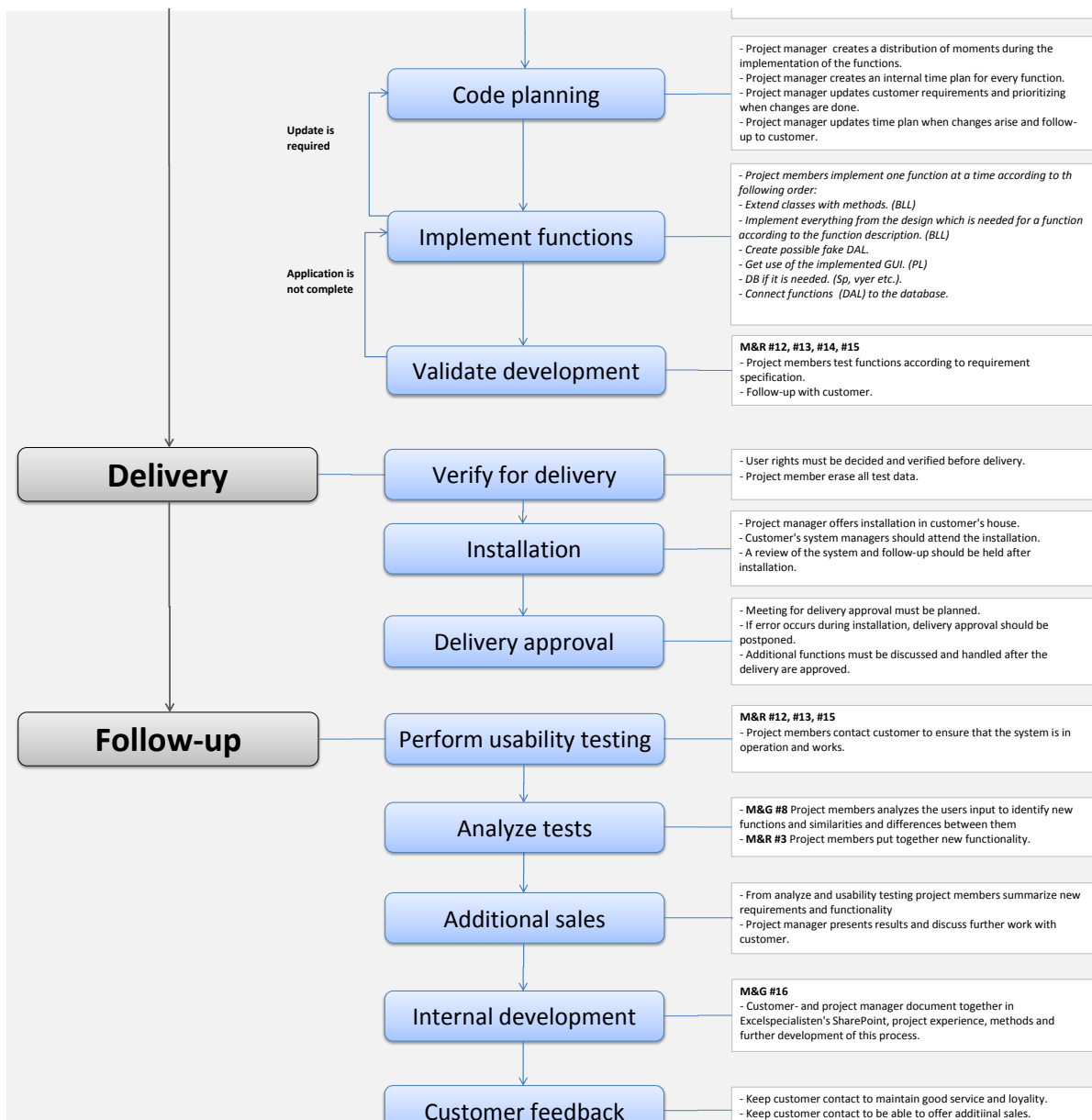
Till att börja med måste processägaren och resten av organisationen aktivt arbeta med att förbättra processen. Förbättringsstrategin fungerar som sådan att man tar bort de moment i processen som inte är värdeskapande och ersätter de aktiviteterna med nya, mer effektiva. I denna nivå ska man också identifiera processens intressenter och uppskatta deras förväntningar. Här tar man också ett viktigt steg i förändringsarbetet genom att kunna visa på att processen ger resultat. Detta gör man med hjälp av mätningar av tid, kostnad och kvalitet.

Öka företagets nivå från 3 till 4 inom metoder handlar om upprepning av nivå 2 till 3. Det som skiljer denna nivå är att det ska adderas ytterligare metoder till de redan existerande. Det ska även finnas utrymme för utveckling av egna metoder så att de anpassas mer för det egna företaget och de specifika projekt de ska behandlas i.

Företaget	Process	Bemanning	Metoder	
Nivå 4	✓	✓	✓	Ständigt förbättrat
Nivå 3	✓	✓	✓	Implementerat
Nivå 2	✓			Definierat
Nivå 1				Sporadiskt

Appendix C





Appendix D

Project Initiation

Appoint project

- Project manager appointed and relevant information about the customer and the project is obtained.
- Decision is taken whether the project is feasible.

Project manning

- Project manager man the project according to complexity, availability and necessary skills.

Preparation

- Preparing for customer meeting.
- **M&G #2** Interview questions.
- *Technical questions.*
- *User centered questions.*
- **M&G #3** Paper prototypes for GUI.

External project

- Present preparatory work.
- **M&G #4** Interview with customer and user.
- Eliciting requirements.

Obtain customer knowledge

- **M&G #5** Summarize collected information.
- **M&G #6, M&R #7** Identify users and describe their situations.
- **M&G #8** Summarize elicited requirements.
- **M&G #9** Analyze users to identify similarities and differences in needs and requirements.

Time planning

- Project manager identifies mile stones and deadlines in calendar time.
- Project manager plan reconciliations for user tests with customer and end users.

Validate customer understanding

- **M&G #10** Prioritize requirements.
- Customer prioritizing requirements as desired.
- Developers prioritizing requirements according to: Most difficult at first, then most useful depending on work time.

Internal project meeting

- Internal meeting with project members, allocating responsibilities.
- Project manager inform developers to create an understanding for customer and project.
- Project manager makes a follow-up with requirement and function description with customer.

Design

Develop user interface

- Project members creates user interface according to requirements and paper prototype as support.
- Implement flows in user interface.
- Inspiration basis is chosen as the colors, logo and customer identification.

Verify user interface

- **M&G #11, #12, #13, #14**
- Project members meet client and present and test the prototype.
- Discuss user interface decisions.
- Present and discuss ideas for future implementation.

Implementation

Create architecture

- Decision about which components the system should consist of, according to prototype:
- Create a flow chart between components according to the requirement specification.
- Identify risks with the chosen architecture.

Develop design

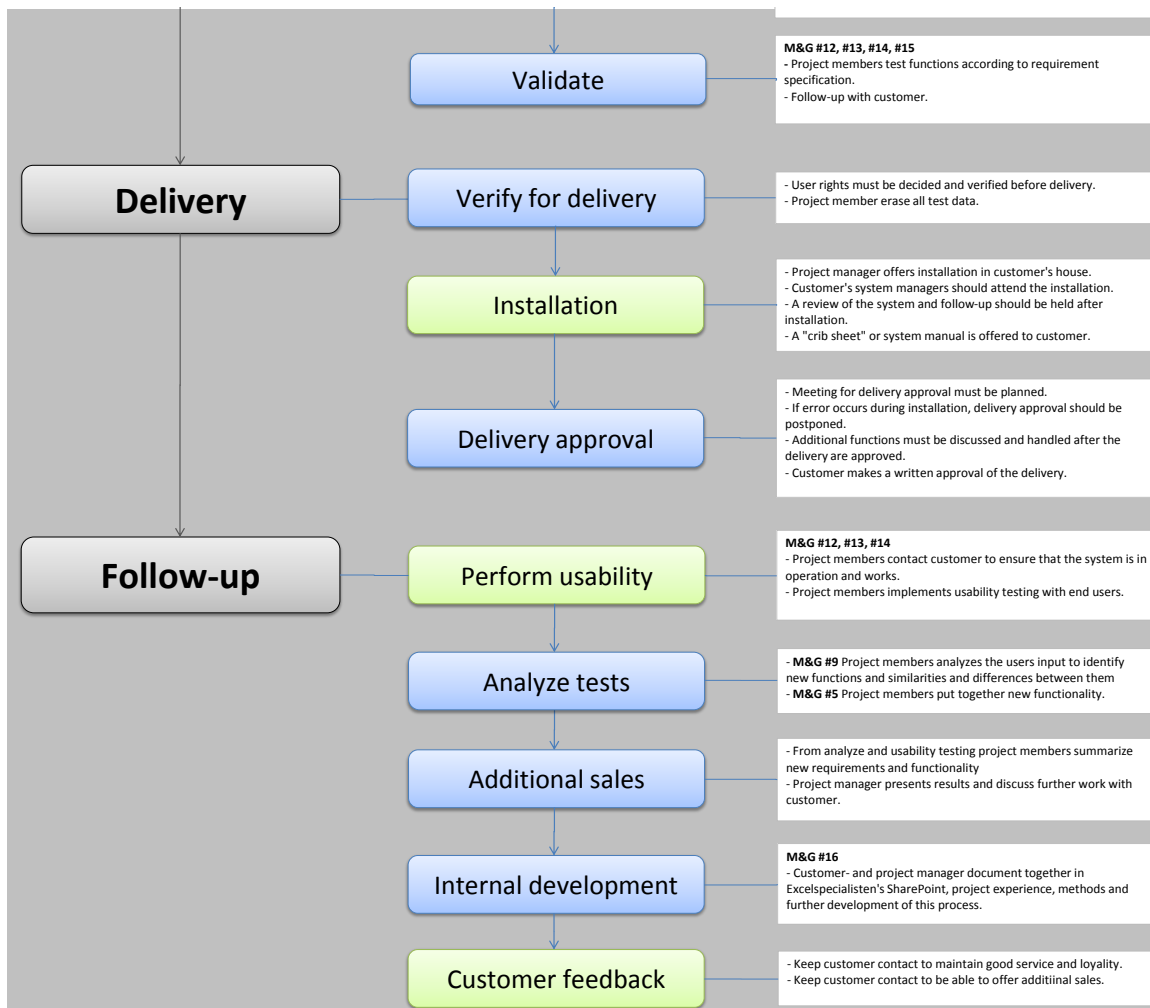
- Overall design:
 - Modules.
 - Public procedures.
 - Classes and types (naming, parametres, returning values).
- The whole function description must be satisfied with this design.

Update project plan

- Project manager creates a distribution of moments during the implementation of the functions.
- Project manager creates an internal time plan for every function.
- Project manager updates customer requirements and prioritizing when changes are done.
- Project manager updates time plan when changes arise and follow-up to customer.

Implement functions

- Project members implement one function at a time according to th following order:
 - Extend classes with methods. (BLL)
 - Implement everything from the design which is needed for a function according to the function description. (BLL)
 - Create possible fake DAL.
 - Get use of the implemented GUI. (PL)
 - DB if it is needed. (Sp, vyer etc.).
 - Connect functions (DAL) to the database.



Appendix E

Meaning unit	Condensed meaning unit Description close to the text	Condensed meaning unit Intepretation of the underlying meaning	Sub-theme	Theme
Krav från Q-matic var: Användarvänligt, lätt att förstå, och automatiserat. Konfigurerat för de olika användarna.	Programmet skulle vara användarvänligt och konfigurerat till de olika användartyperna.	Qmatics Önskan var ett bra och lättanvänt program.	Förväntningar på system	
Ett av Qmatics viktigaste krav var att XLTime skulle motsvara processer de hade tidigare.	xltime skulle motsvara de processer som fanns innan.	Kund förväntar sig att programmet ska fungera snarligt med gammalt system.		
Många önskar sig tillbaka till tiden när man drog ett kort och kom och gick. Men det var komplikationer när man var sjuk och så. Ställs högre krav idag eftersom det ska vara rätt direkt när de ska till löner.	Önskan finns att använda sig av gamla rutiner, men gick inte att rapportera frånvaro. Högre krav ställs idag.	Till viss del vill de ha det som "förr", men med ny funktionalitet.	Intern kommunikation	
man lyssnar bara på vad säljaren säger om slutanvändaren och förutsätter utefter det hur den typiska användaren ser ut. Den kan lätt bli missförstånd.	säljaren förklarar slutanvändaren. Kan bli missförstånd ganska lätt.	Då det inte finns någon direkt kontakt med användare kan det bli missförstånd.		
har hitills inte varit med i den första träffen med kund. Som utvecklare träffar jag projektledaren i första hand som berättar vad som händer under kunds möte.	är aldrig med på första mötet med kund, träffat projektledare istället som tillger uppgifter å krav.	väsentlig information kan förloras eftersom en mellanhand "skapas". Missförstånd kan uppkomma.		
Intern kommunikation har endast gått mellan Niklas och Alex.	intern kommunikation har gått mellan två personer på företaget.	viktigt information som någon bär på allt från säljare till utvecklare kan förloras om de inte är engagerade i utvecklingen.		
Här kan det strula, speciellt när det blir mellanhänder. Man försöker få en förståelse ändå via den kontakt man har. Försöker att undvika detta genom att komma ut till kund och se hur deras befintliga miljö ser ut. Om inte detta går så ringer han och pratar med kund för att få en klarare uppfattning.	problem kan uppkomma vid målgruppsdefiniering pga mellanhänder. Bra att kunna komma ut till kund ist.	många mellanhänderoch sällan kontakt med slutanvändare. Detta kan självklart skapa missförstånd som kommer upp i efterhand.		
kommunikationen skedde via möten under utvecklingen och via telefon efter installation.	kommunikation via möten under utveckling. Kommunikation via telefon efter leverans			
Offerten var redan färdig innan, jag som utvecklare kom in i projektet. Jag kunde inte påverka utfallet av den.	offert redan färdig. Ingen preliminär påverkan på projekt	utvecklare kan missa väsentliga detaljer och förståelse till kund/användare. Säljare kan misseräkna utvecklingstiden.		
målgruppen definieras i ett tidigt skede. I första mötet. Ganska klart.	målgruppen är redan klar när jag kommer i kontakt med projektet.	väsentlig information kan förloras eftersom en mellanhand "skapas". Missförstånd kan uppkomma.		
öppen dialog, vidare kontakt under tiden för att fylla eventuella luckor.	möte - öppen dialog. Luckor fylls med eventuella återkopplingar.	Saker kan missas vid ett öppet möte. Viktiga krav kan bli förlorade		
Tydlighet i krav. Kommunikationsfel. Vissa saker missades mellan xls och qmatic. Henrik känner att han fått inte så snabb respons. Skickade en lista men tog tid att få tillbaka.	Kommunikationsfel, saker har missats mellan parter. Henrik fick inte snabb respons.	Bristfällig kommunikation		
Lite fel i förmedlan till xls. Kommunikation. Kan bero på att vi inte vetat vad vi vill och xls har inte alltid förstått vad qmatic har ordnat.	Qmatic har inte vetat vad de egentligen velat ha, XLS har inte förstått Qmatics behov.	Bristfällig kommunikation.	Extern kommunikation	
		merförsäljning kan minska ifall man inte tar upp de alternativa lösningar kunden kan beställa. Om kunden inte känner till vad som kan utvecklas så kommer de inte veta att de kan beställa det heller.		
Specialanpassningar la vi ingen större vikt på.	ingen fokus på specialanpassade lösningar.			
Andra steg av prototypförslag presenterades internt där applikationen blev minimerad i funktionalitet och anpassad efter Qmatics krav på funktionalitet. Qmatic la till smågrejer som inte ingick i offert.	en andra prototypförslag gjordes internt (2-3 st olika). Endast nödvändig funktionalitet var medtaget nu. Småjusteringar uppkom i efterhand som enligt XLS inte ingick i offert	besluten kommer endast internt vilket kan resultera att kundens bild av applikationen inte finns med i utvecklingen.		
prototyper utförs på lite olika sätt. Oftast ett enda alternativ som presenteras för kund. Applikationer försöker jag göra med papper och penna. Mindre applikationer målar jag upp en bild för mig i huvudet.	prototyper presenteras endast i ett alternativ till kund. Skisser utför jag med papper och penna. Eller i huvudet om appen är av mindre storlek.	kundens önsk mål går förlorade då man väljer att bara visa upp alternativ.		
Det dröjde väldigt länge innan vi fick se web-gränssnittet. Aldrig sett hur det fungerade. Däremot sett excelgränssnittet. I januari, en månad innan drift. Hade velat se det i tid. Fick aldrig se några skisser/prototypes men det hade kanske varit bra i efterhand även om webben var väldigt mkt bättre än det gamla XLTime excel.	Ingen såg webbgränssnittet innan levereans. Såg aldrig skisser eller prototyper. Bra att se i förväg.	Qmatic har inte fått möjlighet att påverka utseendet på gränssnittet.		
funktionsbeskrivningen bygger på att jag har ett antal olika händelser med en given systemaktivitet till. På så sätt ser jag att alltting är kopplat till något, dvs. har en funktionalitet i applikationen.	funktionsbeskrivning målas upp så att alla krav uppfylls.	Alla de krav man har uppfylls. Men saker kan ha missats i tidigare steg.		
Skiljer sig beroende på kund. Om det är en ny kund och han är inte insatt i företaget målar han redan upp en bild vid första kontakt via telefon bland annat. Om han sedan tidigare var insatt i kundens problem behov så kunde han i förväg ta fram och komma på frågor angående krav och utformning på systemet. Men om han har mkt liten kunskap om företaget så valde han att ta fram frågorna under själva mötet.	varierar beroende på kund. Försöker plocka fram bakgrunds fakta om kund för att skapa en bild.	Försöker få fram en bild av kund. Dock måste slutanvändare även tas hänsyn till		
Övergripande beslut av gränssnitt görs i grova drag . Hur arbetet går till idag. Därefter gör man prototyp. Motiveringar fås av kund om det är möjligt. Kommer med förslag.	gränssnittsbeslut i grova drag. Sen gör man en prototyp, kommer med egna förslag	bra ordning att utföra dessa moment dock behövs en mer genomgående förståelse av kund och behov.	Genomförande under process	
Gemensamt val, både från kunder, kollegor och sig själv. Här faller äldre mallar ofta in från min egna kodbank.	designval - både kund och kollegor + erfarenhet.	ser till alla parter. Men motivering behövd inget konsekvent prototyparbete i projekten, dock bra när det görs		
Ibland, och vidare prototyp konstruktion i excel.	skissar ibland annar så utförs prototyper vanligtvis direkt i excel			
Punktlista rakt upp och ner, papper och penna på plats, skissa för att kunna visa kund hur man tänker. Det faller på prototyp direkt.	punktlistar krav . Skissar med papper å penna.	Tidig prototyp vilket är väldigt bra, behövs fler tester för att få fram de rätta kraven bra med mock-up dock behövs en mer genomgående förståelse av kund och behov.		
i 1:a stadiet, mycket prototyp, hur den ska se ut och fungera. Jag gör mock-ups, små prototyper, inget detaljrikt. Gått bättre än vad jag trodde. Gått bra, inte varit några större fel. Folk är nöjda över lag.	gör mock ups. Låg detaljrikedom			
	Nöjda över lag.			
Oftast kund som återkopplar, när de stöter på patrull. Kund får även samtal från oss för att se hur det går för dem.	kund återkopplar till xls vid funderingar. återkoppling från oss med.	bra med återkoppling, någon typ av plan eller standard skulle underlätta för att se hur man ligger till.		
Ingen metod, öppen diskussion. Kommer tillbaka till kontor och skriver 'user stories'. Återkopplar till kund i telefonsamtal för att bekräfta att man har förstått rätt.	öppna möten vid initiering skriver user stories. Återkopplar till kund via telefon.	Viss form av användarfokus men saker kan missas då det inte finns någon standard		
A4, rita fönster och rutor. Som det skulle se ut på skärmbildens innehåll. Flöden ritas även upp som en blandning . Detta fungerar bra på möten.	A4 - ritar mock ups. Målar upp flöden också.	Han gör en low-fidelity prototyp vilket är väldigt bra.		
Det finns lite tanke kring hur vi utformat flöden. Det ska vara lätt gallande ifyllnad av en arbetad vecka och registrera den.	flöden i applikationen har lite baktanke	flöden i applikationen måste ha mkt baktanke då de ska alltid kunna fungera och speciellt i alla de situationer som kan uppkomma.		
Den funktion som används ofta, t.ex. frånvarorarter ligger som default i aktivitets listan.	viss funktionalitet har default lägen.	Bra att de har övervägt dessa tankar, tyder på en viss medvetenhet om att göra användare nöjd.	Genomförande under designfas	
Tar från kundens utseende som logo, färger mm. För att kunna matcha kundens önskemål.	gränssnittet - ser till kundens profil å utseende			
Bra: intuitivt, lätt att använda. Folk fattar hur de ska göra. Lättanvänd	Det som är bra är att det är lätt att använda och att ansättilda vet hur man gör.	Övergripande förståelse för programmet		
Gränssnitt: Tycker det är jättesnyggt och lätt att se saldon mm. Användarvänligt.	jättesnyggt gränssnitt, användarvänligt	Nöjda med gränssnitt		

utgår ifrån egna direktiv och tar designval efter et. Delar av prototyper vara anpassade till flera olika alternativ för att försöka möta kunden på bästa sätt.	designval tas efter erfarenheter och för att möta kund.	Inga genomgående metoder som motiverar besluten.	Omotiverade beslut
övergripandet kommer från bilden av mötet med kunden och den skapta user storyn.	beslut och motiveringar kommer från kundmötet och de skapta user stories.	Ser till vad kunden beskrivit i första hand. Även om det kanske inte är det den behöver.	
Inga input/output krav från Qmatic, mer än att man ska fylla i klockslag och minst ha 30 minuters rast.	input / output är beslutade av XLS förutom 30 min rast och klockslagsformatet.	Kund har lämnat över ansvar på xls, de har inga klara krav	
det beror på kunden och vad för data användning han förser oss med.	prototypen hänger mkt på vad kunden vill ha och vilken info som de ger.	Man gör som kund säger oavsett konsekvens. Kund har inte alltid rätt	
designval, knappar mm. Kommer i första hand från kundens önskan, kan hånda att idéer och erfarenhet presenteras också men det kommer sällan före kundens krav.	uppfyller kundens önskan, presenterar idéer men de kommer sällan före kundens..	Man gör som kund säger oavsett konsekvens. Kund har inte alltid rätt	
krav från kund. Och egna idéer och erfarenheter.	gränssnittet kommer från kundens krav , samt egen erfarenhet	Man gör som kund säger oavsett konsekvens. Kund har inte alltid rätt	
designval bygger jag beroende på kundens krav och utseende.	designval kommer enbart från kundens krav.	Man gör som kund säger oavsett konsekvens. Kund har inte alltid rätt	Hantering av kundkännedom
Att man måste tidrapportera varje dag. Jobbigt när man ska lägga semester till exempel.	Krav att man måste tidrapportera varje dag. jobbigt.	Inte optimerat gränssnitt och funktionalitet	
Tanken med XLTime är att det skall användas dagligen, d.v.s. man tidrapporterar in efter dagens slut.	tanken var daglig användning av xltime	brist på kännedom om användarens vardag och behov.	
Det verkar visa sig att användare går in på XLTime på en mer sällan basis.	xltime används olika beroende på användare.	brist på kännedom om användarens vardag och behov.	
Förväntningar på XLTime var: Projektredovisningssystem tydligt och tidsredovisningen har kommit vid sidan om. Förväntningarna stämmer överens som de trott.	Projektredovisning är tydlig, tidsredovisning har kommit i andra hand. I överlag nöjda.	Fokus på huvuduppgiften försvann.	
Som användare lägger Jenny ungefär 1 minut. Henrik födelar på 7 projekt, ett par minuter varje dag. Tog lite längre tid från början men många har lärt sig snabbt. Inga funderingar på gränssnittet, det är lätt.	Jenny lägger 1 minut, henrik ett par. Tog längre tid i början.	Det tar olika lång tid beroende på vad man ska rapportera.	
Viktiga funktioner: Kom och gick tider. Frånvaro och övertidsarter. Övertidsavlösta: 'kom och gick' får stå kvar. Välja projekt, knappa in och spara De som får betald övertid: Måste även ändra timmar om man skulle jobba	Viktigt: Kom och gick, frånvaro och övertidsarter. Övertidsavlösta ändrar inte 'kom och gick'. Betald övertid kom och gick	Olika användare har behov av olika funktioner.	
En del lägger upp långt i förväg, semester till exempel. De som inte har övertid betald. En del rapporterar så mkt i förväg att det blir fel för att sjukdom uppkommer, då får man gå in och ändra. a) olika ledigheter/frånvaro.	En del tidsrapporterar/registrerar i förväg. Fel uppstår i efterhand.	Olika användare registrerar på olika sätt	
Sen går en del och godkänner i förväg. Föräldraledighet mm.	En del registrerar i förväg	Olika användare registrerar på olika sätt	
8 projekt e nog max för antal projekt. 1 eller 2 är ganska vanligt bland utveckklarna. En del har även 4. Inte så vanligt som för de med lite färre	8 projekt är max, 1-2 är vanligast.	Det finns ett tak för hur många projekt de anställda sitter i.	
Alla typer av användare: vissa grupper tycker att det är svårare. Mycket att hålla koll på i dessa applikationer.	Olika individer uppfattar svårigheter i programmet.	Programmet är inte helt anpassat till användaren.	
Otydligt med datum. Man måste räkna sig fram till vilket datum det är om man vet att man ska fylla i något en speciell dag.	Datum otydliga, måste räkna sig fram	Gränssnittet är inte fulltständigt vilket gör att användaren får tänka själv	Interaktionen mellan användare, system och utvecklare.
Finns dessutom osäkerhet i vilken vecka man är i och svårt att veta hur långt man ska bläddra tillbaka då. Datum hade varit bättre, det har man bättre koll på tyckte användaren.	Vet inte i vilken vecka hon är	datumet är otydliga för användaren.	
Hon använder veckokalendern för att gå tillbaka till aktuell vecka och går till vecka 23. Hon inser att det är fel vecka då hon inte har någon speciell kännedom om vilken vecka det är och går med hjälp av '>' till vecka 24.	Använder kalenderför att navigera. Går till fel vecka.	Gränssnittet visar inte tydligt vart användaren vill navigera	
När hon kommit dit lägger hon till två rader efter varandra av typen restid. Hon ångrar sig här när hon fyller i all restid på en rad och går in på den förstnämnda resetidsrader och säger "Nej, nu gick jag fel", och avbryter. Hon går till kom och gick tabellen och Först ändrade hon en tid till den hon arbetat och sedan klickade hon på ok. Sedan tryckte hon på spara, hon trycker alltid på spara om man ändra ska registrera men hon vet inte när hon behöver det. Sedan klickar hon på 'registrera'.	användare trycker alltid på spara knappen utan kännedom vad den gör för nytta eller inte. Sedan på registrera. Alltid den ordning på flödet.	ingen optimering i gränssnittet.	
Folk glömmar ange övertid som aktivitet, detta är ett problem. Detta bidrar till att en del får ta sin övertid som flex i stället. Man ser inte förrän i nästa lön att det blivit fel och då ordnas detta.	användare glömmar ange övertid, får det som flex ist. Lön ges ut retroaktivt p.g.a. felaktigheter	användare glömmar ange input, vilket bidrar att program generar datan som annan typ	
Han tittar på datumet och inser att det måste vara nästa måndag och klickar på högerpilen. Han får upp rutan med att ändringar gjorts och klickar nej. Han påpekar att rutan kommer upp och vet inte hur han ska hantera den så han trycker bara nej och låtsas	tittar på datumintervall för att lokalisera sig, navigerar med pilar. Blir oförstående till förfrågan och klickar nej. Kommer till nästa datumintervall, ser genom datumintervall att det måste vara måndagen som är den rätta dagen, klickar på	gränssnitt ofullständigt då användare måste tänka själv, feedback från ruta är otydlig	
Han bläddrar fram med pilen och han sparar när det frågas om det under alla veckor för han vet inte vad som är rätt säger han. "Jag känner att detta är jobbigt att det är så många steg"	Han vet inte när han ska spara så han gör det för alla veckor. Han tycker det är jobbigt	feedback från ruta är otydlig om vad användaren ska göra	
Pil höger - nej – ser inga datum räknar i huvudet – mån – tis, rätt d	Räknar i huvudet till rätt datum.	gränssnittet är inte optimalt då man inte ser datum och måste räkna själv vilken dag som gäller	
Klicka på gick rad. Klickar på gick tabell ändrar till 14.30 och klickar	Ändrar gicktid genom att klicka på gickknappen.	De klickar på den rad de verkligen vill ändra på.	
Höger pil – nej, Vecka 28 – klickar på 33.	Först stegvis navigering till vecka, sedan direktnavigering.	personen inser efter en stund att det går snabbare att navigera globalt.	
Den röda knappen är väldigt liten om man ska ta bort projekt.		gränssnitts optimering kan behövas	
Högepil nej flera gånger tills hon kommer till rätt vecka. Kommer till vecka 33. Bläddrar till 34. Klickar på lägg till rad och väljer prjekt. Lägger till 8 timmar på onsdagen och klickar ok.	naviger endast fram genom pil, lägger till rad för projekt inkl tid	trög navigering väldigt mkt klick å läsa.	
Tid 30 sek. klickar på fredagen och klickar direkt på gick-cellen. Sedan går han till spara och registrerar veckan. Bekräftar.	ändrar tid, sparar, registrerar.	Ingen optimering i gränssnittet.	
Hoppar mellan veckor måste finnas. Det finns ju men eftersom han inte kommer ihåg den så borde den synas bättre. Man ser inte att det är en knapp.	användare kommer inte ihåg hur man 'hoppar' mellan veckor.	gränssnittet otydligt	
Ingen tydlig skillnad mellan grönt och svart. Bättre skifta bakgrund än knapp.	otydlig skillnad mellan färger	Gränssnittet behöver ändras	
Otydliga punkter. Man ser inte att detta är en knapp att trycka på att fylla i.	Otydliga knappar för kom och gick.	gränssnittet otydligt	
Många registrerar av misstag. Redan på måndagen, eftersom man hamnar direkt på veckan som är så kan detta hända. Då måste Jenny läsa upp för att de ska kunna justera de felaktigheter. Ofta att läsa upp.	Registrering av misstag. Administratör måste läsa upp. Händer kontinuerligt.	Det är lätt att göra fel. Gränssnittet är otydligt.	Hantering av gränssnitt

Sjuk-lägger till en rad. Men flyter inte i dag och tid. Missar detta.	Sjukfrånvarande missar att lägga till tider.	Man vet inte hur man ska fylla i sjukfrånvaro
Förstår inte varför jag får en fråga när angående om jag vill spara ändringar när jag byter vecka. Jag har ju inte gjort några ändringar.	Får en fråga om att "spara ändringar" när jag byter vecka, har inte gjort några ändringar.	Användaren blir förvirrad då den tvingas ta ställning till om den vill spara eller inte trots att den aktivt inte gjort någon förändring.
Klickar höger , nej höger nej. Vänster nej,. Vänster nej. Vänster nej för att komma till rätt vecka. Räknar till torsdag. Jag skulle vilja ha datum på dagarna. Klickar på kom, och ändrar på gicktiden. Klickar 18.30 och sedan ok. Lägger till en ny rag, inarbetad tid. Känner inte igen sig. Avbryter. Klickar igen avbryter. Tar bort raden genom att klicka på röda krysset. Lägger till ny rad. Över tid i kompid. Klickar på torsdagen rad och klickar starttid. Skriver 1800. Klickar ok.	Räknar i huvudet till rätt datum. Utför resterande steg för att slutföra uppgiften	gränssnittet är inte optimalt då man inte ser datum och måste räkna själv vilken dag som gäller
Klickar högerpil. Det måste finnas något annat sätt att navigera med men klickar på högerpil. Och sparar inte för hon vet inte hur. 10 repetitioner av högerpil nej till rätt vecka. Går till den valda dagen onsdag och ska avbryter. Lägger till ny rad. Scrollar ner och hittar projektet. Klickar ok. Klickar på onsdagen för projektet. Anger 8 timmar och OK.	klickar väldigt många ggr för att ta sig fram med pilarna. Känner att det kanske finns något annat sätt att navigera på men vet inte hur. Utför resterande steg i uppgifr	hon vet inte riktigt hur hon ska kunna navigera på ett bättre sätt än detta även fast hon känner att det borde finnas något sätt att göra det på
Knapparna sitter väldigt nära, spara, registrerat.Får popup men många gör felet ändå. Kan bero på att de sitter fel.	Knappar sitter väldigt nära. Får popup om registrering men fel görs.	Knappar kan vara felplacerade
Klickar på högerpil. Klickar nej här för jag har inte gjort några ändringar. Räknar dagarna för att hitta rätt datum. Klickar på gickraden. Ändrar till 14.30. trycker ok.	Räknar i huvudet till rätt datum.	gränssnittet är inte optimalt då man inte ser datum och måste räkna själv vilken dag som gäller
Börjar med att spara. Klickar högerpil och klickar sedan på vecka och väljer v 37. Klickar ok. Lägger till rad på onsdagen och scrollar ner. Väljer thbru. Lägger till 8 h på onsdagen och sparar.	ser till att tidigare steg är sparade. Utför uppgiften och sparar sitt utförande	kontrollbehov att spara hela tiden både på gått å ont.
Klickar på torsdag kom och gick. Ändrar sina tider 0700-1730. Lägger till rad och klickar på aktivitetkod. Klickar på frånvaro. Klickar på övertid. Klickar på restid och OK. Klickar på filen för restid och ändrar sluttiden till 0800. Lägger till en ny rad och klickar på övertid och restid, sedan ok. Lägger till starttid 16.30 klickar ok. Sparar sedan.	navigerar runt för att hitta restid. När han väl hittar det skapar han två stycken rader för att ange morgon å kvälls resa. Sparar när han har utfört sina steg	vet inte var han ska hitta. När han väl utfört uppgift så sparar han givetvis.
Räknar fram till tisdag. Lägger till en rad. Går till frånvaro. Väljer uttag komp. Klickar på tisdagen fyller. Tid 2.01 blir konstigt input. Fyller sedan i att hon jobbar till 14.30. Blir fel med kompen och hon går tillbaka och tittar. Hon väljer att inte spara.	lägger till rad för att utföra uppgift, vid input sker inputsproblematik . Kompen visar inte korrekt när hon anser att hon är klar	input är inte optimal då man uppenbarligen gör fel som ger ordentlig konsekvens. Uttövers så är inte uppgiften slutförts helt korrekt.
Klickar sig bakåt till vecka 24, han lägger till ett godtyckligt projekt för att utföra hos kund 4.45. Sedan händer ingenting han vet inte hur han ska gå till våga alls för att lägga till restid. 50 sekunder går då han inte vet vad han ska göra och hittar sedan restid i övertid. Hon bläddrar fram till 33 med hjälp av att klicka på veckoknappen. Sedan fyller hon direkt i vilken vecka hon vill gå till, vecka 33, tryckte på ok. Sen lade hon till en ny rad. Valde det projekt hon skulle välja med en gång och väljer denna. Hon påpekade här att Hon tittar på veckans datum för att identifiera intervallet. Byter vecka ">", klicka på ja i förfrågan om osparad status. Räknar från måndag som hon ser är 21 till 22.. 23, identifierar onsdag som den gällande dagen. Klickar 'gick' för vald dag, klickar på vänsterdelen	användare uppfattar osäkerhet i sin uppgift och stannar ett antal ggr.	flödet / gränssnittet är inte klart nog för att kund ska förstå sin uppgift.
Bugg i funktionalitet när man trycker enter vid inmatning av tid.	Klickar på den specificerade veckan och lägger till ett projekt plus den tid hon jobbat. Känns som många steg	Flödet är för komplicerat/ej optimerat
Jag använder ofta enter, jag föredrar att använda tangentbordet.	tittar på datumintervall för att lokalisera sig, navigerar till rätt intervall, räknar i huvudet fram till rätt dag.	gränssnitt ofullständigt då användare måste tänka själv
Ologiskt att inte kunna fylla i tider för morgon och kväll utan att skapa två rader.	Bugg i funktionalitet vid entertryck	Känd bugg i funktiolitet som inte är planerad att åtgärdas.
Jag vill ha ett smartare sätt att navigera mellan veckorna förutom vänster och högerpilarna. Det kanske finns något sätt men jag vet inte hur man gör.	Föredrar att använda tangentbordsinput.	Användarna använder inte programmet på samma sätt som det är tänkt att användas.
Jag saknar att inte kunna lägga till flera kom och gick tider under samma dag. Nu är jag tvungen att lura systemet och det förhåller sig inte till verkligheten.	Ologiskt att behöva skapa två rader för restid.	Programmets funktionalitet är inte uppbyggt som användare har tänkt sig.
Användande: Inne varje dag, en del varje vecka. Skiljer sig. Beror på hur man jobbar. Om man jobbar som vanligt utan projekt och utan övertid godkänner ju bara. De fyller bara i frånvaro. De som jobbar i projekt går in varje dag för att fördela sina timmar i olika projekt. Väldigt olika hur de rapporterar. Henrik går in varje dag för att inte glömma vad han har gjort men andra gör det en gång i veckan.	Måste kunna navigera på ett annat sätt än med höger- och vänsterpil.	Användaren missa funktionalitet i programmet. Använder det inte pga att det är okänt/syns inte för användaren.
	Saknar att inte kunna lägga till flera kom och gick tider under samma dag.	Programmets funktionalitet är inte uppbyggt som användare har tänkt sig.
	Personer utan projekt godkänner bara rapportering, Personer i projekt redovisar dagligen. Olika rutiner för hur de rapporterar tid.	Olika användare använder programmet på olika sätt
Förde mus över torsdagens rader men gör inget. Klickar på lägg till ... scrollar ner på projekt scrollar upp. Klickar på aktivitetsdropdown menu – klickar av den. Scrollar på projekt igen.	letar efter restid i olika menyer.	personen vet inte var den ska hitta det specifikt söka
Går till övertid klickar på restid klickar på ok – klickar på ok . tänker anger på sluttid 09.00. Klicakar på ok. Lägger till en ny rad på övertid igen klickar ok. Klickar på den nya raden anger 16:30 som starttid 17:30 slut ok.	Lägger till restid i två rader.	utför samma sak 2 ggr för att slutföra uppgiftens krav, kan uppfattas som väldigt onödigt
För musen över projektet och klickar sedan på lägg till rad. Väljer övertid. Och klickar på restid och OK. Går till arbetstid. Klickar på kom, ändrar till sju. Klickar ok, klickar på gick och lägger till 17:30. klickar ok. Gåår till restid och klickar. Klickar på sluttid. Ändrar till 0900. Klickar ok.	utför restidsuppgiften som hon uppfattar den är uppbyggd genom att "lura" systemet.	hon utför uppgiften felaktigt genom att lura systemet. Det kommer fungera men det är itne snyggt vid rapportering
För musen över måndagen och sedan trycker vänststerpil. Klickar nej vid popup. Höger 2 ggr- klickar nej varje gång.	Navigerar stegvis till veckan. Svarar nej på förfårgan om spara ändringar.	meddelandet kan uppfattas jobbigt då användare egentligen intr utfört någon ändring.
Klickar på torsdagen och ändrar till 0700-1730. Lägger till ny rad och övertid. Klickar restid och ok. Klickar på rad för restid på torsdagen och ändrar tiden till 0700-0800. Lägger till ytterligare en rad och klickar på torsdagen för restid och lägger till 16.30-1730. Klickar ok.	Lägger till restid i två rader. En för morgon och en för kväll och justerar sin arbetstid så den stämmer överens.	utför samma sak 2 ggr för att slutföra uppgiftens krav, kan uppfattas som väldigt onödigt
Bugg i funktionalitet när man trycker enter vid inmatning av tid.	Bugg i funtionalitet vid entertryck	Känd bugg i funktiolitet som inte är planerad att åtgärdas.
När man dessutom trycker enter för att konfirmera projekt så kommer "lägg en rad" upp.	Lägg till en rad kommer upp vid entertryck.	Ett flöde som användare är vana vid fungerar inte
Fungerar inte så bra i firefox. Folk använder olika webbläsare på företaget.	Fungerar inte bra i Firefox, Olika webbläsare används.	Begränsat program
Om man jobbar halvdag så blir det konstigt med rasten efter som det står att man måste ta 30 min rast även när man inte jobbat så mkt. Istället så fifflar han med rapporteringen och fyller i att han jobbat en halvtimme extra.	Konstigt med raster då man ej jobbar så mycket. Man får lura systemet.	Inte tillräckligt med funktionalitet för att denna inmatning ska fungera.
Onödig feedback vid bytet av vecka. Det är det första som händer på måndag morgon när man ska fylla i förra veckans. Helst få bort frågan då man inte gjort en ändring. Men även en cancel. Så man kan kolla om man verkligen ändrat något.	Onödig feedback vid byte av vecka.	Bristfälligt i gränssnittet.

Hantering av funktionalitet

Produkten känns lite ofärdig. Ologiska saker händer i systemet. Justera någon tidbank till exempel.	Ofärdig produkt, ologiska situationer uppkommer.	Teknisk del ofullständig, generar oförutsedda fel.	
Systemet fungerar sådär, vissa saker har inte gått importera men det är nog bara tillfälligt. Men de mesta som skett är lätt att reparera.	Systemet fungerar inte som det ska. Problem som uppkommit har kunnat repareras.	Programmet är inte fullständigt.	
Använda tangentbordet mer.	Vill använda tangentbordet för navigering och input.	bristfällig inputsmöjligheter	
Dåligt: XLTime tänker inte, tar emot alla inputs. Mer kontroller. Ibland vill de lägga till två rader. Att de jobbar i två skift. Lägga till i två skift när de kanske jobbar hemifrån.	XLTime tänker inte, tar emot de flesta inputs. Fler kontroller. Vill lägga till flera rader för kom och gick.	XLTime är inte smart nog.	
det varierar mycket och det är ofta väldigt svårt att definiera det eftersom det finns både olika typer av målgrupper. Genom att ha kännedom om bakgrundsförståelsen, målar jag upp en bild av den typiska användaren.	svårt att definiera målgrupp. Enklare om man har viss bakgrundförståelse till vad applikationen ska användas till.	Bakgrundsförståelsen kommer endast från inbäringsnötet, inte från djupare kunskap om användare	Metodik vid kravhantering
Man har inte använt några user tests. Man har gått på egen erfarenhet och känsla.	inga usability tests, egen erfarenhet istället.	XLS grundar inte sina beslut på något.	
Första prototypförslaget som kund kom i kontakt med var identiskt till det "gamla" XLTime.	prototyp visning nummer 1 - var originalutseendet. Onödigt info och funktionalitet som inte skulle användas fanns där.	risk för förvirring. Då kunden får se ett utdrag som ändå inte kommer vara aktuellt till slutanvändaren.	Metodik vid prototyp
Det fanns nackdelar med att vi presenterade detta då mycket av funktionalitet som den innehöll skulle ändå tas bort, blev väldigt mycket onödigt fakta.			
Tanken var att skapa ett system som var lätt att använda med funktionalitet som prio. Men det blev ont om tid och gränssnittet följde med per automatik utifrån det som hade skapats från tidigt skede.	lätt system med funktionalitet som prio. Gränssnitt följde med per automatik.	gränssnittet får ingen tid att testas, utvecklas och skapas på ett naturligt sätt. Om det bortkastas finns hög risk att användaren inte känner sig bekväm i det eller att flöden blir onaturliga mm...	
efter leverans.. Kundensvar mer fel kommer fram med rätt data. Förvaltning & återkoppling lämnas över.	kundens ansvar och testa och tycka själv.	man släpper system till kund utan att göra kontrollerade tester.	
leverar till kund och låter kund testa. Men det har även hänt att jag varit ute hos kund och hjälp de starta och lära sig applikationen, detta dock på kunds begäran.	kund gör tester hos sig. Ibland har man varit ute hos kund och hjälpt de "komma igång".	Lämnar över till kund och låter de själva testa. Kan missas viktig fakta.	
Återkopplar till kund. Det beror på kundens önskemål. En del vill prototypen för tidig testning medan andra vill ha en ganska färdig prototyp.	återkopplar till kund när det är levererat för att se hur det går. Olika kunder olika situationer.	Är inte med vid leverans. Gör inga tester.	Metodik vid leverans
lämnas i princip bara till kund. Med viss återkoppling.	vid leverans lämnas app till kund. Återkoppling sker vid senare tillfälle.	man släpper system till kund utan att göra kontrollerade tester.	
efterarbetet hänger ihop med att vi återkopplar efter ungefär en månad och ser hur det går för kunden. Även vid senare tillfällen såsom 6 månader senare. Ifall det är småsaker som ska läggas till så kör man inte igenom kundens profil, men om det är större förändringar så gör de det.	återkopplar efter en månad vanligtvis och sedan i ett jämnt intervall för att se om nya saker ska implementeras.	kund kan ha glömt av fel som dykt upp om inte tester görs omgående.	
Hon kan inte se andras kommentarer i attest.	information finns inte tillgänglig	gränssnittet är inte fullständigt	
Hon kan inte se klockslagen för övertid. Hon ser endast hur många timmar den har varit.			
Under tidens gång utfördes små justeringar, kund har aldrig haft Restiden var krånglig. En lathund till alla funktioner så att man kan se hur man ska göra vid nya saker.	information finns inte tillgänglig småjusteringar utfördes under tidens gång. inga krav på krånglig inmatning, inte logiskt, lathund skulle underlätta	gränssnittet är inte fullständigt varken kund eller utvecklare har en gränssnittet otydligt	
Feedback från användare: Engelsk version. Summering av projekt. Har jag jobbat tillräckligt. Syns däremot i attestering.	Vill ha engelsk version och summering av timmar i projekt.	Gränssnittet/Funktionalitet är inte fullständigt än.	Potentiella kompletteringar
Svårt att se att fördelat alla timmarna. Men varit svårt tt se tydligen eftersom vi inte varit medvetna om att	Logga ut och summering finns numera. Svårt att se totala fördelade timmar i projekt.	Gränssnittet/funktionalitet är inte fullständigt.	
Något slags default för hur man jobbar i projekt, nu får man göra allt varje dag. Vore bra om man kunde skriva in direkt hur många timmar man jobbar i ett projekt. Bara om man vill lägga till en som det är viktigt.	Ett default för tider i projekt.	Funktionalitet fattas.	
'Kom och gick' måste kunna läggas till flera gånger. Det finns användare som jobbar sina åtta timmar i skift under dagen.	Användare jobbar i skift. Måste lägga till kom och gick två gånger	Finns funktionalitet som efterfrågas	
En del vill hellre tabba sig mellan cellerna istället för att få upp en popup för varje dag. Detta skulle minska många extra klick.	Tabba sig fram mellan celler för att minska klick	Användaren vill ha flöden som inte finns i nuläget.	