



UNIVERSITY OF GOTHENBURG

A Case Study: Moving from ad hoc to agile software development

MARKUS EDSTRÖM

Bachelor of Science Thesis

**Report No. 2009:027
ISSN: 1651-4769**

A Case Study: Moving from ad hoc to agile software development

Author: Markus Edström <edstromm@ituniv.se>

Software Engineering & Management, IT University of Göteborg

Abstract

A small, young and rapidly growing software company will be investigated throughout this paper using the interpretive case study method. The company has desiderated support in structuring daily work, contemporaneously the company has an objective; desires of following agile development principles. The company has realized that a change is necessary to become a functional company, producing high quality software. The goal is to help the company with the transition between their ad hoc approaches towards agile development. To meet the company wishes, this paper presents a brief walkthrough of some agile best practices. It also describes how different interview techniques and belonging findings and results may help the company see the problem from a new perspective. By using the media synchronicity theory, the company's communication channels are identified and analyzed. The repertory grid technique helps the company identifying their big spawn of various customer types. The results from both interviews will be taken under consideration while discussing how the company can start following agile work practices and what must be done for a successful transition from ad hoc to agile development. At the end of this study it is supposed that the existing gap between theoretical models and actual ways of performing work is reduced and that the company can use provided material being this paper and findings from the collaboration during their transition period on their transformation road towards agility.

Keywords

Agile, ad hoc, repertory grid, media synchronicity theory, change, scrum, extreme programming, lean development, dynamic system development method, case study, toolbox.

1. Introduction

Defining, changing, refining, mitigating or adapting to software processes has always been and will always be an activity for people involved in the software industry. It doesn't really matter what of the mentioned characteristics that are involved, it's always a challenge. New or old to the branch doesn't matter either. As an organization, project manager or development team, you still have to understand your software process, why you use it, how you benefit from it and the strengths of it. As in any other branch there are related trends. When the major actors of a branch try new things, everybody else also wants to do it. Finding defects in a software process challenges you to either change process or invent your own innovative option. This can be something completely new or it can be as simple as modifying something that already exists. Independent from the age of the software process methodologies, companies struggle and still have problems with finding the suitable one.

As Nerur et al [1] describes the challenges of migrating from traditional to agile methodologies in his article, attributes such as fundamental assumptions, control, management style, role assignment, communication, customers role and organizational form are inspected. The purpose of this case study is to help an organization (which from now on will be referenced as "Beta Software") with similar challenges that Nerur describes, but instead of migrating from traditional to agile development, the author of this article will describe similar challenges but in sense of moving from ad hoc to agile development. The change itself in this case has many perspectives and similarities with Nerur such as communication, documentation, customer relations etc. Other suitable articles will be used as references in order to understand the many dimensions that influence a change process. This paper will ergo focus on a certain company, called Beta Software and its transition from their ad hoc way of developing software towards agile software development.

The purpose of this research is to understand the challenges that Beta Software encountered when moving to agile software development. This challenge will be described throughout a case study. The case study is interpretive and will be combined with several other data collection techniques e.g. active participation, informal question time, semiformal interviews and formal interviews. By active participation the researcher will understand the company culture and through informal question time and semi formal interviews the researcher will understand what the project managers do and how they interpret the software process at Beta Software. By interviewing project managers using the media synchronicity theory [13] and repertory grid technique [14], the researcher will understand the communication channels at Beta Software and the organizations various customer types.

Performing this study will educate and inform Beta Software with the basic knowledge of some agile methodologies. The goal is to make the organization understand what agile development is about, and how they can choose a best practices suitable for their situation. Performing this study will help Beta Software in the transition to agile, understanding the benefits of it and to eliminate the gap between theoretical models and how work actually is executed. The study will also contribute with material to a virtual toolbox where the project managers can share documents, experiences, knowledge and relations. This toolbox will also be used for communication, feedback looping and reporting. By evaluating the communication channels and the customer characteristics at Beta Software, recommendations will be provided in order for the organization managing their different customer projects in an agile way. A very brief walkthrough of four agile methodologies; scrum, extreme programming (XP), lean development (lean) and dynamic system development methods (DSDM) will be provided, enabling both readers of this document and the organization to understand some fundamentals that agile methodologies contain.

After the related research section and the data collection results presentation a discussion will describe concrete examples that will help Beta Software on their road of agility. These examples are based on previous related research and the results from the data collection.

2. Related research

In the beginning of this section, a brief summary of an article by Nerur et al [1] will be given in order to understand what this paper and study builds on. Together with this summary, some agile methodologies e.g. Scrum, extreme programming (XP), lean development (lean) and dynamic systems development method (DSDM) will be described to provide the reader and Beta Software with some basic agile knowledge. Understanding agile methodologies and the challenges of adopting them will help the reader follow the data collection and discussion part of this article. The purpose of this walkthrough is not only to give the reader an understanding of agile methodologies. When this collaboration between the external researcher and Beta Software has ended, Beta Software will be able to use the report itself as a knowledge reference while considering using an agile method in some project or finding particular interested in a certain one. Possessing shallow knowledge will help the organization in digging deeper and developing agile skills in their change process.

2.1 Summary of past research done by Nerur

Following text is a summary of previous research done by Nerur et al [1].

Changing technologies requires innovative software development methodologies. The traditional plan driven software methodologies lack the abilities of flexibility in matters of dynamic adjustments in the development process. Unlike traditional software development, the agile methodology claim being more effectively in sense of incremental software development. Some developments methods e.g. extreme programming, scrum and feature-driven development fall into the agile category. The objective of the article is to highlight the challenges of using popular agile methodologies instead of following the principles of traditional development. While using agile methodologies, companies must understand strengths and weaknesses of the agile principles. Past research reveals that its impossible to replace current tools and technologies and therefore changing software development process is a complex organizational challenge. It is therefore important to understand these methodologies while changing an organization. The agile methodologies have some common characteristics e.g. fundamental assumptions, control, management style, knowledge management, role assignment, communication, and customer's role, project cycle, development model, desired organizational form/structure and technology. Agile methodologies also find light documentation, responsibility and collaboration as important factors while developing software. Understanding the described characteristics and important factors visualizes the challenges of making agile methodologies work. As the agile methodologies rely on people in difference to traditional development that relies on processes, the challenges in sense of management, collaboration, culture and other human factors are though. "Agile methodologies are ideal for projects that exhibit high variability in tasks (because of changing requirements), in the capabilities of people, and in the technology being used" [1:78]. "They are also appropriate for projects where the value of the product to be delivered is very important to customers" [1:78]. "Organizations must carefully asses their readiness before trending the path of agility" [1:78].

Beta Software has many interesting similarities with what this article describes. As the organization stands upon change, factors such as control, management style, knowledge management and role assignment are interesting to take under consideration for this unique case. With this article as a starting point, the researcher has been able to decide how interviews should be formed, where extra effort and investigation should be done and what the challenges would look like for Beta Software. The author extends and adapts previous work done by Nerur [1] to suit Beta Software's situation as accurate as possible.

2.2 Scrum

As in the sport rugby, scrum development also focuses on a team of individuals acting together, moving forward, working tight and integrating. Understanding your role is an important task of collaborating in a scrum team. Focus and priorities must be clear to the team [2].

Iterative & Incremental - Scrum is an iterative and incremental development method. This means that the development team goes back to completed tasks or activities and either change, improve or add new things [2].

Pregame, Game, Postgame - Pregame, game and postgame are three major phases of scrum development. In the pregame phase, planning is done together with system architecture and high-level

design. During the game phase the sprints (iterations) are concerned. This is where the actual analysis, design, development and engineering are done. Closure is an activity of the last phase, where the project is delivered to the customer [3].

Daily scrum - Daily scrum is a short meeting held each day where the group discusses what has been done since last meeting, what has to be done until next meeting and potential problems and other issues that has to be solved. The scrum master (group leader) is in charge of these short meetings that have duration of approximately 15 minutes [4].

Backlog - Two types of backlogs exist in scrum development. The product backlog lists all necessary elements needed to deliver a final product and the sprint backlog lists elements taken from the product backlog that should be completed during a sprint (iteration) [4].

Develop, Wrap, Adjust & Review - As the header describes, the four attributes: develop, wrap, adjust and review are activities connected to the sprints (iterations) in scrum development [3].

Flexible schedule - A deliverable may be required sooner or later than was planned from the beginning. This means that the project members also have to think, work and act flexible [3].

Small teams - Scrum development teams are small, containing three to six developers. Developers, documenters and quality control staff form these teams. There can be one or more teams working at the same project [3].

2.3 Extreme programming

This section will describe some major characteristics for the software development called extreme programming. Throughout this section, extreme programming will be referred as XP. As any other agile software development method, XP has its own unique characteristics. Some of these will be described in the text below.

Small releases - XP has two different views of the concept called small releases. In the first view, the software is released, run and tested. It also delivers business value chosen by the customer. This happens during each iteration. What the customer does with provided software is totally up to them, but usually they evaluate or release it to the end users. Providing visible software to the customer keeps it open and tangible. In the second view the XP teams also release their software to the end users as well [5].

Pair programming - Two programmers sit side by side at the same machine while writing code. By doing this all code is reviewed by at least one other programmer [5].

Simple design - An XP team follows design suited for the current functionality of the system. No work is wasted, and the software is ready for being further developed [5].

Test-driven development - To understand feedback XP uses test driven development. By either programmer testing or unit testing, it's ensured that the code is working and of high quality. The cycles of adding tests on existing code are short [5].

Customer tests - To demonstrate that a feature is working, the programmers perform acceptance tests on themselves and on the customer. This proves that the functionality is implemented in a correct way [5].

Coding standard - By following a coding standard, developed code will look as if a single person wrote it [5].

Collective ownership - Any programmer-pair can improve or change any code at any time during the project. By doing this, all code can get many people's attention which leads to better code and less bugs [5].

2.4 Lean development

This section will provide a short description of lean software development. Some of the bullet points are lean principles, other typical lean characteristics.

Eliminate waste - Eliminating waste sounds very rough, but basically everything not adding value to the customer is considered as waste. This includes: unnecessary code and functionality, delay in the software development process, unclear requirements, bureaucracy and slow internal communication [6].

Amplify learning - By learning from past experiences, solved problems should be documented to avoid waste when reconstructing the same solution when meeting the same problem in the future [6].

Decide, as late as possible - By deferring commitment in sense of hard or impossible decisions, decisions that are hard to change should be moved to absolutely last minute. There shall also be a backup plan if the decision of choice does not work [6].

Muda - Any activity that requires resources but does not contribute to any output value is considered as a waste in lean software development, called muda [7].

Re-use - By reusing of components enables improved quality and reduced cost since the components being reused already have been developed, tested and approved [6].

Value - Value is what the customer thinks is useful or valuable to him [6].

QFD - QFD means quality function development. This technique helps teams in the translation of customer requirements [6].

2.5 DSDM

Dynamic systems development method (DSDM) is a framework for rapid application development (RAD) [22]. The DSDM consortium maintains DSDM [8]. Since a consortium develops the framework the research from the academic world is poor. In this report the nine principles of DSDM will be presented to meet the company's desires.

Active user involvement is imperative - By involving a small crowd of users throughout the project effectively reduces errors in terms of user perception that reduces the costs of errors [9].

Teams must be empowered to make decisions - According to DSDM authorization between participants and managers slow projects down. Therefore participants should be given limited authorization in relation to some project related activities. The authorization includes: requirements in practice, which functionality needs to be in a given increment, prioritization of requirements and features, refining details of the technical solution. By empowering the participants with this authorities, time spend on communication and requests can be lowered [9].

Focus on frequent delivery - By delivering frequent errors can be detected quickly and easy to reverse. This comprises both program code and documentation [9].

Fitness for business in criterion for accepted deliverables - The DSDM framework recommends focus on delivering software that meets the business needs and that enhancements should be done during later iterations [9].

Iterative and incremental development is mandatory - To keep complex projects manageable; projects must be divided into small feature packages. Each release presents new features until all business requirements are met. DSDM means that smaller increments make it possible to change requirements [9].

All changes during development must be reversible - As DSDM advises iteration through small increments they mean that loss of work during change phases is very limited [9].

Requirements are base lined at high-level - Some high-level requirements are needed to limit the degree of how requirements can change during the project lifecycle [9].

Testing is integrated throughout the lifecycle - DSDM requires testing in the early stages of the development process. Control groups can tests elements such as interview documents [9].

Collaborative and cooperative approach - DSDM provokes collaboration between technical and business staff in a project to get a trustworthy and honestly environment. This enables easy requirements collection and honest feedback of the product [9].

3. Research method

This section describes how the researcher of the paper conducted a case study at Beta Software. The role of the researcher will be described together with a description in what way this case study was conducted and explanations that will help the reader understand why it was done in this particular way. The section also provides the reader with a background, problem definition and some goals. Related research has been done using three main sources; IEEE Explorer [17], The ACM Portal [18] and Google Scholar [19].

3.1 Background

Beta Software is a development company in the software industry. The company provides the market with services that are configured or implemented to meet each unique customers requirements and wishes. Throughout a formal encounter collaboration between Beta Software and the IT University of Gothenburg has been established. By establishing this connection Beta Software hopes to receive help, advices and material that supports them in their daily work, that will be described more detailed in the sections problem definition and goals. Beta Software shall also be able to advance provided material. By collaboration it's also supposed that this bachelor thesis is written to provide best service possible to Beta Software and a contribution to the academic world and anyone interested in agile software development.

3.2 Problem definition

Beta Software suffers from three main problems. Identifying and describing these are the starting point of this bachelor thesis. To start with the organization expresses a growing gap between documented theoretical models such as software process and actual ways of performing work. There have been unsuccessful attempts in diminishing this gap and the organization need to solve the problem before it becomes too large and unmanageable. This is followed by the desires of running agile projects, but the lack of knowledge is preventing the organization of doing so. By compiling an adaptive toolbox, the organization hopes that the project managers will be supported in their daily work, that the described gap will shrink and that agile methodologies can be used in different projects. Beta Software suffers from material and ideas how this shall be accomplished. Beta Software feels that these three main problems has been put aside for to long and is now requesting help in overcoming them. As the organization uses an ad hoc way of developing software, the problems mentioned must be solved before Beta Software can present them as an organization working agile.

3.3 Goal

By collaborating with Beta Software, there are several goals. Solving their defined problems is prioritization one. As the collaboration started, Beta Software has shown an open-minded state. They desire critical monitoring, research and external opinions by a person unknown to their organization culture, work area and product library. The goal is to study the organizations theoretical models, understand how they function and at the same time identify suitable agile best work practices that may support the project managers. By performing research at the organization and dig into agile methodologies, the goal is to provide material that can work as a starting point while compiling an adaptive toolbox and start the change process from ad hoc to agile development. The goal of the collaboration is twofold: [A] Support Beta Software with the three described problems in the problem definition, [B] Contribute with an innovative and interesting report to the academic world and agile community.

3.4 Case study background

At this point of the paper, concepts such as traditional development [1], agile development [20] and ad hoc development have been mentioned. This small section will help the reader understand the differences between these three concepts and their meaning, relation to each other and how they should be considered for this unique paper. When reading about traditional development in this paper, the reader should think of traditional software process and their belonging activities while developing

software. A traditional development process is e.g. the waterfall model where the projects moves from a starting phase through various other phases, ending in product delivery. The traditional way of developing software does not include iterating over tasks and elements.

Agile software development [4] is concerned with iterating over phases, stakeholder integration and is very different compared to traditional software development. E.g. agile software development doesn't put the same focus on documentation as traditional software development.

As this paper extends Nerurs [1] previous work where he describe a transition from the mentioned traditional software development to agile software development this paper uses the same idea that slightly has been modified. Instead of looking on the transition between traditional to agile software development, this study looks at Beta Software's' unique way of developing software, referred to as *ad hoc* development throughout this paper and their transition from ad hoc to agile software development. This paper will also focus on how the organization can start following concepts from the agile school and transform from an organization without structured ways of working, to an organization following agile principles.

Beta Software is experiencing a growing gap between their theoretical models and ways of working. On the same time they don't follow any structured ways of working, they rather use a combination while solving problems and developing according to the employees various knowledge. The organization also has many types of customers, which require specialized ways of encountering. Beta Software has grown very fast in the last couple of years and poor time has been spent on defining structured ways of developing, reporting, collaboration and so on. The organization has worked a lot by trial and error, trying to find a reasonable way of developing software. As the organization feels that this does not work anymore, they have desiderated support while looking on this issue. Therefore this paper focuses on the transition from the described old ad hoc way of developing towards agile development. The organization desires to follow agile principles, and the goal is to help the organization to understand what agile software development is about, how to adapt it and how to leave the old ad hoc way beyond them.

3.5 Case study

To collect data, understanding the results and making a strong analysis the researcher will execute this investigation by following the principles of an interpretive case study [10]. The investigation is a social enquiry performed on humans working in a technical environment. By following the interpretive case study method, the researcher will be able to parse the employees' interpretations of the world that they call their workplace. Following subsections will describe how and why this case study was executed in this particular way.

3.6 Researchers role

As Walsham [11] advocates the role of the researcher should not be changed during the time of the investigation. The researchers role in this case study is based upon the problem description provided by Beta Software, being under investigation. This researcher will critically monitor the project managers in the organization. To understand their daily duties, management style and ways of solving problems, the researcher will act as an involved researcher but still an outsider [11]. By taking the role as an active participant [11], the researcher will understand how Beta Software functions. From this stage the researcher can make decisions of how to interpret the organization and in what way to collect various data. The goal is to physically attend at the office as often as possible. In this way the researcher will be able to put him into the organization and understand how it functions. Simultaneously as the organization is being investigated, the researcher documents and presents the results in a paper, being this article.

Beta Software basically advocates help in three main areas; the growing gap between theoretical models and actual ways of performing work, problems in the transition towards agile development and plans of creating a toolbox for the project managers. This toolbox would contain support material for the project managers and the abilities of storing and sharing information, material and knowledge within the organization.

Reducing the described gap means spending time on research within the organization. Different types of interviews, where the results will be saved, analyzed and finally followed by suggestions, will

accomplish this. The goal is to use collected data and use it in a matter that can function as material for a toolbox and at the same time reduce the described gap. As the organization also wants to work according to agile principles, this paper will mainly address agile methodologies. As some agile methodologies are briefly presented in this paper, it is Beta Software's duty to spend time on further elaboration and research in that area.

As the collaboration ends, it is totally up to Beta Software if they want to follow directives or try other options. If Beta Software decides to follow provided directives, time must be spent on further research and elaboration around identified findings done by the researcher. Due the timeframe of the collaboration it is not possible for the researcher to follow Beta Software from problem definition to the end of an eventual transition.

3.7 Case study execution

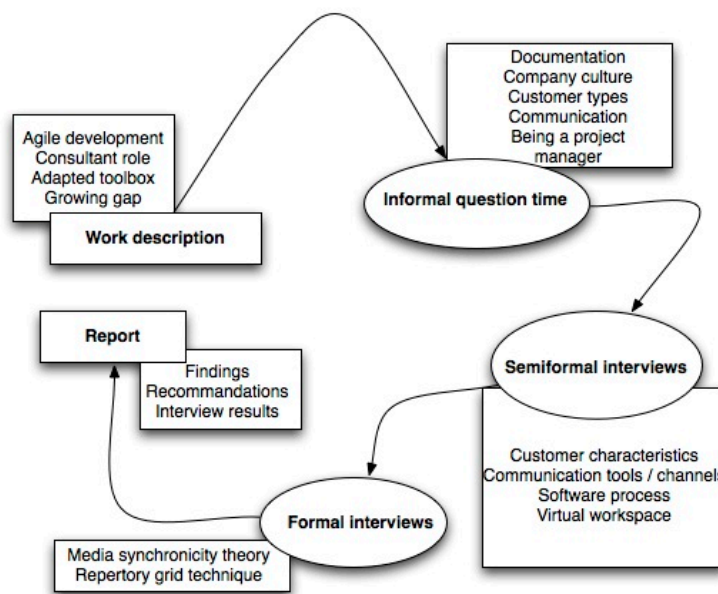


Figure 1. Visualization of various steps involved in performing the case study.

The above drawn figure represents how this case study will be executed. By studying it, one can understand that there are several phases with connected activities. As the time frame of the research project is pretty short, a structured way of working must be followed to make all involved parts aware of what is going on, what has been done and what is left of the collaboration.

As the study moved from a work description and a problem definition, it was followed by informal question time with Beta Software's project managers. This has helped the researcher to understand the company's culture, their documentation, how they communicate, their customer types and the responsibilities of a project manager. Some findings were made such as; the organization uses light weight documentation, they have an open-friendly relation among them and the project managers main duties is interacting with customers in different projects. The project managers also use a virtual workspace called "SharePoint" [12] to share information regarding their projects. The founding's of the light documentation claim are based upon the organizations old "Wiki" [21], and project manager monitoring. The researcher did not find tons of documents on the old wiki, and it seemed like the project managers only managed a couple documents for each customer project such as; requirement specifications, weekly reports and project plans. In the beginning of the case study it was quite hard to extract usable information from the informal question time. Asking project mangers about their role in the company was a hard task in the beginning. The extracted information from these sessions was poor and not much of use. This is probably because the project managers felt inconvenient and threatened in the beginning of the study. Opening your mind and talking about your role is sensitive and might be interpreted as one intrudes on their work description and position in the organization. As time went by,

informal question time became more valuable the end, when a personal relation containing trust was build up between researcher and project managers.

With the previous described findings, the researcher was able to setup semiformal interviews with the project managers. During these sessions, the project managers were asked to draw their interpretation of the software process in the organization. This task opened up their minds, where they could discuss pros and cons with the software process, pros and cons of their ad hoc ways of working, together with some wishes describing a desire of following agile principles. The software process map was done a long time ago, poorly updated and hard to follow. The main problem of following it was the different customers. One project did not look the other alike, and in most cases the software process had to be modified in one way or another to suit the various projects. Unfortunately the software process interpretations looked very similar to the already documented material. This is probably because of the existing imprinted picture that exists on the project manager's minds. Since they expressed difficulties in redoing the visualization, they most have a limited idea of how the process looks and could look like. This should not be classed as something negative, rather positive as it gives the researcher the opportunity to criticize the software process and present new suggestions and terms while treating the topic. The organization itself is familiar with the agile concept. They have managed a couple of scrum projects and been using some iterative activities in sense of planning and feedback loops from customers. Furthermore it feels like the organization rather needs to be educated in the agile topic before they are ready to run agile projects. Finding suitable agile best work practices that can be combined in a project would be a great activity for the organization, rather than focusing on the statement "we must work more agile". As the organization seems open minded to change and willing to learn, this should not be considered as a problem, rather as a challenge.

During the semi formal interviews the project managers also had to describe what different tools that are used for communicating within the organization. They were also asked to contribute with their personal interpretations of different customer characteristics. In the daily work the project managers use communication links such as e-mail, instant messaging clients, telephone and mobile phone communication as well as virtual meetings. This means that the project managers have to be familiar with various communication tools for different customers regarding of technical skills and preferred communication channels. As the organization sells their solutions it is important to be able of adapting to customer claims. At the same time as the communication channels were documented, the project managers listed several attributes that would form the structure of the formal interviews that later will be described. By letting the project managers identify the customer characteristics, the supportive material for the interview is totally compiled by the organization to suit their situation and current state. The results from the characteristics identification were mixed. As some project managers listen same attributes, doublets had to be removed. Some attributes such as "customer far away" and distributed customer had to be merged, since they basically have the same meaning.

During the semiformal interviews, the project managers became more open and willing to discuss their issues and describe their thoughts of how change could benefit them. Establishing this personal relation with the project managers made upcoming parts of this case study easier in sense of discussions, feedback and so on and so forth. At this stage, the semiformal interviews were not only about extracting information. As the project managers became more open, two-way communication started together with discussions criticizing existing ways of working together with improvement ideas.

Choosing interview techniques can be tricky. The formal interviews are based upon identified communication tools/channels and customer characteristics. The researcher set up two formal interviews, one following the media synchronicity theory [13], where communication channels are evaluated and one interview where customer profiles are mapped. The second interview was performed according to the repertory grid technique [14], supported by an Internet tool, Web Grid IV [15]. The repertory grid technique is a technique that makes it possible to compare basically anything. As an example, the word light can be listed, and it's opposite, dark. Than one can compare different environments such as; school home, work, church, cinema and disco on a scale 1 to 5, where 1 is light and 5 is dark. In the same way the attributes for the customers will be compared and documented in the data results section. Its worth mentioning that both formal interviews were performed in a way to suit the organizations communication channels and tools together with their unique projects shaped in different ways depending of the type of customer.

The reason for doing these different types of interviews is to help the project managers with their main daily duties, communication and customer interaction. The goal of the interviews is to enable a

discussion were communication channels can be analyzed and improved together with a basic understanding of various customer types. By Understanding these two mentioned artifacts, once again the related research play a great role in the transformation process. These interviews also started wild discussions of how to implement an adaptive toolbox and what it concrete should contain. As the project managers slowly started to understand in what way they were led as they were interviewed, they had a lot of suggestions and thoughts of the toolbox content. They also became able to discuss in what different ways the toolbox should work as a support tool and in what way of forming it to make it most useful.

4. Data results

This section will present the results gathered from the earlier described elements of the case study. It will also describe how the techniques were used. Furthermore the results will be discussed how the data can benefit Beta Software to eliminate the described problems in the problem areas and how they can be further used as this study has ended.

4.1 Semiformal interview results

By executing semiformal interviews, a common interpretation of the current software process has been enabled to visualize. As one can se it is pretty much following the waterfall model, except for the one iteration between implementation and testing.

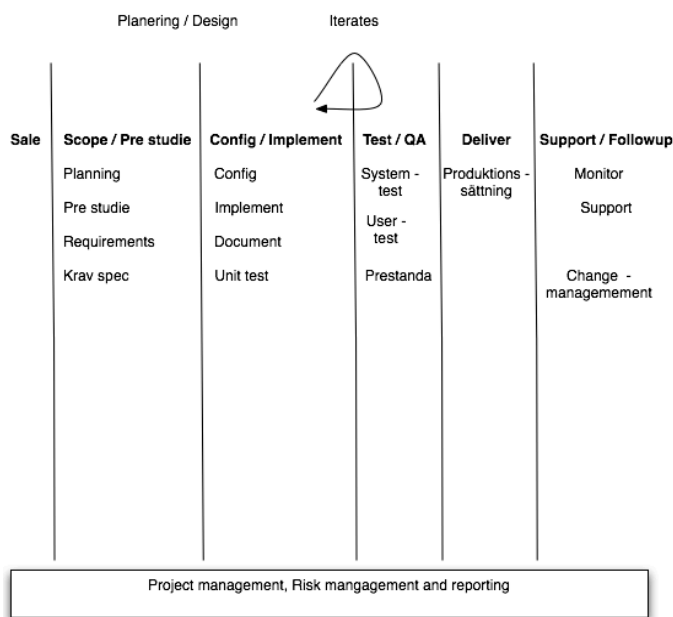


Figure 2. Common compiled apprehension of used software process.

This process map was compiled in order to maintain a discussion what to put in a toolbox. As we can see a typical project has different phases. The first idea was to put different information according to the various phases while creating a toolbox. Beta Software provides their customers have a wide product supply. The organization can develop new software or modify existing software to suit a certain customer. The first and last vertical lines in the image show pre and post activities. The activities in the middle are always thought of while developing a product. The organization has different project forms e.g. proof of concept projects where the organization proves their skills to a customer, development projects where a product is delivered. The organization can also offer maintenance and support services. Establishing new customer relations means that Beta Software must prove a winning concept and maturity while producing software to their customers. Its therefore suggested that the old software process model is reworked or that considerations are done to present other material proofing that the company has a structured way of working. The second statement is most certain the most suitable one, since many customers will not understand a software process map, or that they actually valuable the

delivered product more than an explanation of how to accomplish this. How this result was processed will be described more in detail in the discussion section.

4.2 Formal interview – Media synchronicity theory

The first formal interview was executed to interpret one of the main focuses described by the project managers, communication. The article “Managing team interpersonal process through technology: A task – technology fit perspective” [13] describe various communication channels while collaborating on a common task, in the form of virtual teams. This study has a strong connection to the article, in sense of communication channels and working as teams. The original canvas of communication channels has been slightly modified to fit the study, but the different attributes belonging to each way of communicating is unchanged. In order to understand what the attributes mean, read following quoted explanation, written by Likoebe [13:977]. “*Immediacy of feedback essentially captures the synchronicity of the medium, whereas symbol variety speaks to the availability of multiple cues and language variety that are supported by the medium. Parallelism captures the possibility that some media permit multiple simultaneous conversations, and rehears-ability represents the ease with communication can be rehearsed and edited prior their transmittal. Finally reprocess-ability embeds the ability of medium to maintain a history or memory of the communication that has occurred.* “

| FACE TO FACE | 1 | 2 | 3 | 4 | 5 | EMAIL | 1 | 2 | 3 | 4 | 5 |
|--------------------------|---|---|---|---|---|------------------------|---|---|---|---|---|
| Immediacy of feedback | | | | | x | Immediacy of feedback | | | x | | |
| Symbol Variety | | | | | x | Symbol Variety | | | x | | |
| Parallelism | | x | | | | Parallelism | | | | | x |
| Rehearsability | | | x | | | Rehearsability | | | | | x |
| Reprocessability | | | x | | | Reprocessability | | | | | x |
| SHARE POINT | 1 | 2 | 3 | 4 | 5 | NET MEETING | 1 | 2 | 3 | 4 | 5 |
| Immediacy of feedback | | x | | | | Immediacy of feedback | | | | x | |
| Symbol Variety | | x | | | | Symbol Variety | | | | x | |
| Parallelism | | | | x | | Parallelism | | x | | | |
| Rehearsability | | | | x | | Rehearsability | | x | | | |
| Reprocessability | | | | | x | Reprocessability | | x | | | |
| TELEPHONE 1 - 1 | 1 | 2 | 3 | 4 | 5 | TELEPHONE X - X | 1 | 2 | 3 | 4 | 5 |
| Immediacy of feedback | | | | | x | Immediacy of feedback | | | x | | |
| Symbol Variety | | | x | | | Symbol Variety | | | x | | |
| Parallelism | | x | | | | Parallelism | | x | | | |
| Rehearsability | | | x | | | Rehearsability | | | x | | |
| Reprocessability | | x | | | | Reprocessability | | x | | | |
| INSTANT MESSAGING | 1 | 2 | 3 | 4 | 5 | | | | | | |
| Immediacy of feedback | | | | | x | | | | | | |
| Symbol Variety | | | x | | | | | | | | |
| Parallelism | | | | x | | | | | | | |
| Rehearsability | | | x | | | | | | | | |
| Reprocessability | | | | x | | | | | | | |

Figure 3. Interview results from media synchronicity theory

The presented data above is based on the identified communication channels expressed by the project managers during a semi formal interview. The table shows the communication channels that also are rated on a scale one to five. The interview was performed on four project managers that gave different answers throughout the interviews. The collected data was documented and merged together. The figure above shows a merged arithmetic value.

By merging the results, two kinds of discussions can be held. The project managers and the researcher can start by discussing the merged values and what they really mean to the organization while communicating internally and externally. Followed by this the project managers can discuss and compare their individual results among them. By discussing the individual results they can compare pros and cons, describe their interpretations of certain values while integrating either with a customer or with a college. By studying the results, Beta Software will also be able to put effort in improving certain communication channels to make them stronger and make more usage of them. As the project managers expressed that communication looks the same independent on internal and external communication it is important to discuss this as well. Is it so that the organization is too formal internally, or not formal enough while collaborating with their stakeholders? With the data results in mind, the organization will be able to start a change process communication wise. There are also opportunities of measuring the communication flow against agile methodologies or changing ways of

communicating, as was desired and seen as a problem when this study started. The results from this interview may also be integrated with the adapted toolbox that later will be compiled. The results of this semi formal interview will also be further elaborated in the discussions section.

4.3 Formal interview – Repertory grid technique

The second formal interview was executed to interpret the second main focus of the project managers, customer projects. Upcoming data collection has been gathered by using “the repertory grid technique” [14]. This technique helps the researcher in mapping up different customer types and characteristics for the study. As the data is sensitive for the organization that wishes to stay anonymous, modifications has been done while visualizing the results from the interviews. To avoid exposing the organizations customer projects, the original customer names have been replaced with the letter C and a belonging number. Each C plus a number represents a real customer. The thirteen identified constructs (e.g. weak process – strong process) have been compiled by the project managers. This data collection source is the largest one for this study. The data presentation figures have been done by using the web tool “Web Grid IV” [15]. The results will be presented in forms of different images. Since the data collection embraces a large amount of data, it is important to visualize it in different way to being able to analyze it in a fair way. The goal of this data collection is to map up different customer types and characteristics involved in Beta Software’s everyday work. The results will be used while discussing the toolbox concept, measurements against the media synchronicity theory will be done and comparisons against agile methodologies will be performed. Following figures will present the data in different way and come with an explanation. During the interview, three additional comment sections were added as gantry for the discussion that took place at the company. These three sections had to be answered how it affected: “planning“, “execution” and “steering directions for getting a smooth running project”. As the time consuming interviews were held, they were also recorded on a computer making it possible for the researcher to go back an both listen and reflect over answers that were either hard to understand or preferably interesting for the study. This interview was the most time consuming activity during the collaboration between Beta Software and the researcher. Four managers attended the interview, and as one can see a total of thirteen projects were rated, analyzed and discussed.

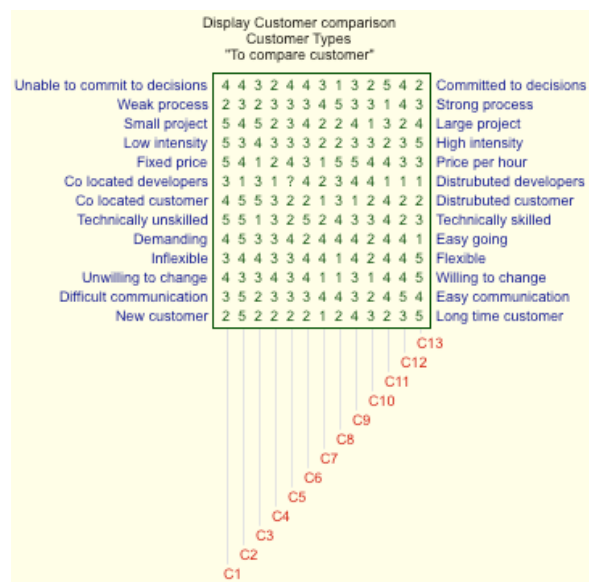


Figure 4. A table showing output result from an interview performed by using repertory grid technique.

This figure visualizes all customers (elements) involved in the study and all constructs used for the interview. The figure is a basic table that gives an overall representation of the collected data. While discussing changes in the organization, the outputted data from the interviews can easily be fetched. When this data was presented for the organization, confusion arose. The table is useful when an employee at Beta Software has to look back on how a certain project is identified on a special construct. By saving the table the opportunity of adding constructs and elements also exist for further

extensions or work on the customer characteristics. As the table was presented, it was of high importance that the project managers understood to be able to follow upcoming data representation types. This first table forms the base for the upcoming representation types.

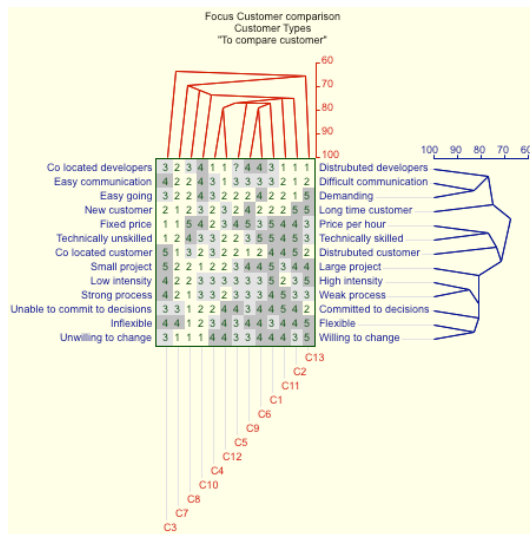


Figure 5. A focus cluster showing output result from an interview performed by using repertory grid technique.

As previous figure, this one also presents the output data in a table. This type of visualization is called a focus cluster. The focus cluster has a scale that runs from the value 100 to the value 60 for both the customer (elements) and the constructs. This helps the viewer to distinguish between similar elements and constructs. By mapping up the similarities a discussion can follow in order to understand how the results have affected both elements and constructs. Clustering the values and identifying similarities between constructs and elements can hold many types of discussion among the project managers. The first type of comparison is an individual comparison for each project managers and his unique projects. E.g. the project manager of project c4 and c12 can see that the top scale connects, meaning that there are strong connection and similarities between the both customers / projects. As each project manager understands the relation between his projects, he will be mature to collaborate with other project managers where similarities are identified between their projects. The deeper on the scale one looks, the more projects will be involved and thereby a higher number of project managers have to attend the discussion. By having these discussion the project managers will be able to discuss similarities, dissimilarities, pros, cons and even coach each other and change experience. Together with the first figure, a new customer project can be rated and there after compared. Doing this will help the organization while choosing development method, ways of managing the project, creating reporting structures, choosing communication channels as well as understanding the customer, enabling a smooth project as possible. As this advantage of the focus cluster was presented to the project managers, wild discussions started and a brain storming session was planned.



Figure 6. A pringrid map showing output result from an interview performed by using repertory grid technique.

This graph visualized the different construct attributes and puts the customers closest to the attribute that was ranked highest for the customer. Take the customer “C9” as an example. It was rated with a 5 on the construct price per hour and rated with a 4 on the construct easy going, as the figure visualizes. This is just another way of presenting the results, but in forms of a graph. This way of presenting data can be easier for some persons to understand than looking at numbers in a table. The pingrid can be used to give a quick overview to spot typical characteristics for certain customers. Even though it was not as widely use as the two previous described tables, it should be presented and explained. As one can see, there are many different ways of presenting the fetched data from this interview and the only thing that stops the creativity is the mind of the people using this images.

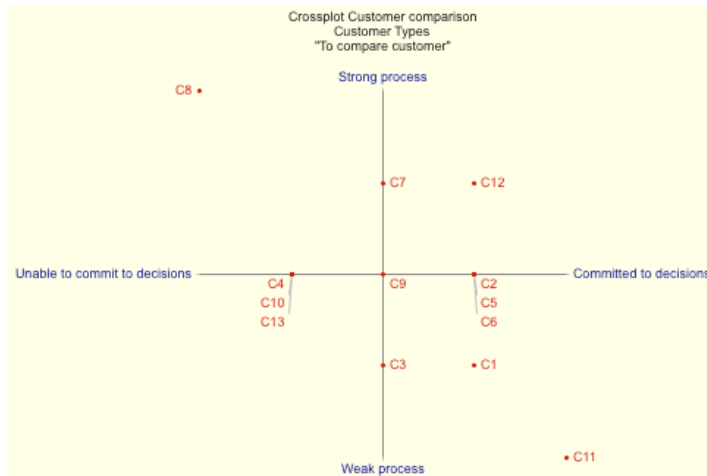


Figure 7. A cross plot showing output result from an interview performed by using repertory grid technique.

This is another graph representing the data. When compiling this graph, the user decides what to constructs that should be compared. As the image shows, customer “C9” is in the center of the graph meaning it had the value 3 on both constructs. Customer “C8” in the top left corner has the value 5 on the strong process construct that indicates the vertical value with 5 on top, and the value 1 on the construct unable to commit to decisions that indicates the horizontal value with 1 on the left side. By analyzing the customer against construct with the previous image, discussions were held in order to manage a customer, communication wise, process wise and other methods used for this study.

5. Discussion

When reading this part of the paper it is assumed that the reader understands the reason of the case study, how it was performed, the results from the data collection and some basic knowledge of four different agile methodologies. This discussion will be divided in four main parts. In the first part the customers and constructs are discussed in relation to the media synchronicity theory. In the second part the author elaborates on the toolbox concept and how this should be planned for further extension. In the third part a discussion will be held around the organizations change process toward agility, the transformation from ad hoc to agile software development and also extend Neur et al’s [1] previous research. Some agile principles from the related research chapter will also be discussed in relation to Beta Software’s’ situation and how the organization could make use of these. In the last and fourth part contributions and achievements will be discussed serving as verification that all goals have been met and that the organization has received the material that was requested in the start of this collaboration together will a discussion how this paper contributes to the academic and agile society.

5.1 Customer characteristics and constructs evaluation combined with media synchronicity theory and agile methodologies.

This part of the discussion will elaborate around how the results from the media synchronicity and repertory grid technique interviews will be useful for Beta Software. Further will these two data result sources be discussed and concatenated with some agile methodologies. To make it easy for the reader to follow the discussion, two customers will be considered and compared. By doing this, the study shows how strong the techniques are while analyzing these customers, independent of how differing the results may be.

Before starting an elaboration and discussion, let us carefully study the following images and have it in mind while reading upcoming sections. Study the values of both customer C1 and customer C2. Take extra notice of the framed constructs and belonging results. These symbolize the major differences between these two customers. The discussion will mainly be held around the differences of these two customers.

Let us also study the results from the media synchronicity theory interview. These results will also be discussed in relation to the two customers and agile principles.

| | | |
|-------------------------------|-----|------------------------|
| Unable to commit to decisions | 4 3 | Committed to decisions |
| Weak process | 3 2 | Strong process |
| Small project | 4 5 | Large project |
| Low intensity | 3 4 | High intensity |
| Fixed price | 4 1 | Price per hour |
| Co located developers | 1 3 | Distributed developers |
| Co located customer | 5 5 | Distributed customer |
| Technically unskilled | 5 1 | Technically skilled |
| Demanding | 5 3 | Easy going |
| Inflexible | 4 4 | Flexible |
| Unwilling to change | 3 3 | Willing to change |
| Difficult communication | 5 2 | Easy communication |
| New customer | 5 2 | Long time customer |
| | | C2 |
| | | C1 |

| EMAIL | 1 | 2 | 3 | 4 | 5 |
|-----------------------|---|---|---|---|---|
| Immediacy of feedback | | x | | | |
| Symbol Variety | | | x | | |
| Parallelism | | | | | x |
| Rehearsability | | | | | x |
| Reprocessability | | | | | x |
| NET MEETING | 1 | 2 | 3 | 4 | 5 |
| Immediacy of feedback | | | | x | |
| Symbol Variety | | | x | | |
| Parallelism | x | | | | |
| Rehearsability | x | | | | |
| Reprocessability | x | | | | |
| TELEPHONE 1 - 1 | 1 | 2 | 3 | 4 | 5 |
| Immediacy of feedback | | | | | x |
| Symbol Variety | | | x | | |
| Parallelism | x | | | | |
| Rehearsability | | | x | | |
| Reprocessability | x | | | | |
| INSTANT MESSAGING | 1 | 2 | 3 | 4 | 5 |
| Immediacy of feedback | | | | x | |
| Symbol Variety | | | x | | |
| Parallelism | | | | x | |
| Rehearsability | | | x | | |
| Reprocessability | | | | | x |

Figures 8 & 9. Customer C1 & C2 comparison and media synchronicity data results analysis.

5.1.1 Customer C1

Starting from the top of the cluster table, the image reveals that customer C1 is a price per hour project with co located developers and also a distributed customer. The customer is technically skilled and easy going. The customer is also easy to communicate with and a long time customer.

Now that we know some attributes of customer C1, we can start discussing them deeper, one by one and elaborate how it affects Beta Software.

As customer C1 is a price per hour project, it will be difficult to have a budget for the project. At the same time its important to make achievements and progress while working on the project. The project managers must make estimations on how many hours they can afford spending on various tasks and how often they can meet their customer. They will also have to make decisions in the project, where to put the focus. They cannot afford spending time on correcting faults that will become too expensive for them. The customer will most certainly not be interested of paying Beta Software for correcting mistakes caused by them. The developers on the project are co-located, meaning that the project manager has his team close. The customer is distributed and this means that the project managers must travel to meet the customer physically. The customer is technically skilled, being able to understand documentation, progress reports and terms, concepts and difficulties connected to the project. The customer is easy going, that means that they trust Beta Software in what they do, are flexible in sense of schedule change and probably positive to change that can benefit the project even more. The communication with the customer is easy, meaning that they are easy to get hold of, they give valuable response and are able to use various communication sources. The last attribute tells us that the customer is an old customer, where collaboration has existed before. This is probably connected with the attribute easy going.

As these assumptions have been made, how can we measure and combine them with the agile material and communication channels analysis that has been made? Let us look on some key words from the agile methodology scrum, described in the related research section and see how it can suit this particular case.

As the customer is technically skilled, they will be able to use some kind of virtual tool for scrum projects where backlogs are presented. By doing this the customer can see the progress on different tasks, see how many hours that is planned for different activities and feel safe that they are paying the right amount of money to Beta Software. They can also verify that the company is fair while estimating how many hours either a task will take. This means that Beta Software can run an iterative and incremental project that the customer is able to understand and monitor. Working as a small team will also benefit the project. It is easier for a group of tops six developers to travel together with their project manager, compared to twenty people. This will make it possible for the development team to be present at both headquarter and by the customer when needed. The customer, being an easygoing customer, enables a flexible schedule. They will probably put more value on the delivered product than complain on a slightly delay. Daily scrum meetings can be held. The project manager will hold these daily scrums with his development team, and the customer is also able to sit in if he so wishes, virtually. Mentioned suggestions follow agile guidelines described as “customer collaboration over contract negotiation” [20]

The above text might sound to good to be true. How is it possible for the project manager to communicate with both development team and customer being distributed? Will all concerned actors understand? How will discussions be stored, understood, prepared and processed?

As we now have looked on customer C1: s attributes, it abilities to follow a scrum project, it is time to look on how the communication flow could look like. Since analysis has been made and a data result for this exists, it will also be taken under consideration.

By using e-mail communication, Beta Software will cover parallelism, rehearse-ability and reprocess-ability. For fast feedback, the development team or the customer can be reached by telephone. In order to supply a high symbol variety, net meetings can be held. As the customer was described as a technically skilled customer, it is assumed that they will be able to follow and use previous described communication channels and concerned tools.

5.1.2 Customer C2

Starting from the top of the cluster table, the image reveals that customer C2 is a fixed price project. The developers are not in the same building as Beta Software, but not hundred miles away. The customer on the other hand is distributed and at the same time technically unskilled. Customer C2 is not demanding, neither easy going. The communication is difficult and it’s a new customer.

What really differs customer C2 from customer C1 is that customer C2 has a fixed priced project, a new customer being technically unskilled a bit more demanding and hard to communicate with. Thinking of this tells us that this project does not have the same preconditions as the project with customer C1. Therefore other conclusions must be made and the same agile key words might not be suitable in this case.

As the project is fixed price, Beta Software must make a more detailed plan. It might also be necessary to put more effort on the project than calculated for in the end. Neither the development team nor the customer are co – located, meaning that plans has to be done while arranging meetings with both parts. As the customer is technically unskilled he wont be able to understand as much vocabulary, documentation or activities compared to customer C1. Since the result shows the number one, on the scale one to five, the customer might have big problems of understanding what is going on at all. The customer is not demanding, nor easy going. This means that the project manager will not be possible to take own initiatives, and that it might be necessary to convince the customer that certain changes are required. As the customer is new to Beta Software, the organization must proof their competence and build up trust and confidence to the customer.

As we understand these assumptions, it might appear that customer C2 is not as manageable as customer C1. A scrum project would be hard for this customer to follow; therefore other options must be considered and presented to this customer enabling a successful project and collaboration. One should not see this as a problem, rather as a challenge and an opportunity. The challenge is to find an agile work practice that will work for this customer and the opportunity is trying something different and new.

As a small research has been made upon the related research section of this paper, it doesn't seem that is possible to strictly follow one agile principle for this customer. A combination of several agile methodologies requires from Beta Software's side to run this project.

As the customer is technically unskilled and has limited understandings in sense of software terms and development, he is not able to manage requirements specifications, architecture or technical terms. Therefore it is suggested that the organization presents small deliveries and focuses on frequent delivery. These two terms are taken from extreme programming and dynamic system development method. As the customer is new and the organization might wish to establish further collaboration, it is crucial that Beta Software shows their expertise in the area and that the customer builds up trust in them. By customer tests the customer can see what they are buying and accept the product. Their opinion and what they value will be important while moving on in the development process. The two keywords customer tests and value are also agile, taken from extreme programming and lean development. It is worth mentioning once more the low technical skills of the customer, which can be met by eliminating waste. Activities, requirements or functions that does not provide value to the project should be eliminated. The customer should only get what they really need and are paying for. Fancy special functionality and advanced preferences are totally insignificant in this case. As the project is fixed priced, Beta Software should also consider re using old components, saving time and cost on this project. This concept is also taken from the lean development library.

How can Beta Software accomplish and communicate these several best practices in a good way to a remote customer? Once again, the media synchronicity theory helps us out and gives value to the concerned parts of this project.

Beta Software expresses that immediacy of feedback is highly rated on face-to-face [Figure 6] communication. The main interested for the organization should be receiving feedback from the customer, as it is technically unskilled and pretty demanding. The other attributes such as rehearse-ability and reprocess ability are not that important, as improvisation most certainly must be made when the customer is unable to understand. Workshops where customer and Beta Software collaborate could be the key to a successful project.

As we have looked on two different customers with different projects we can understand the wide spawn of differences, challenges and opportunities for Beta Software. As this report provides to concrete examples, it is Beta Software's duty to do similar investigation on their current projects and make deeper research in agile development. By doing this they will slowly start to understand what their typical customers look like, what agile principles are most suitable for the various types and how communication should be established.

5.1.3 Construct comparison

As customer C1 and C2 has been analyzed, following text will give a similar description that instead focuses on the constructs e.g. (weak process – strong process).

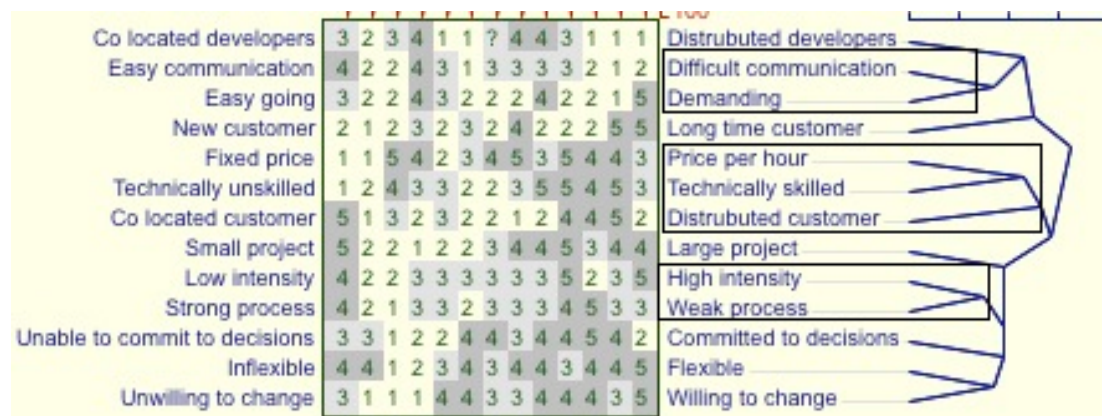


Figure 10. Constructs comparison

As an example, look at the two attributes high intensity and weak projects. They are not only two attributes that interconnect and belong to each other; there is a reason for this, the answers from the managers during the interviews. Several questions can be asked as a starting point for a discussion around the attributes, questions such as; is the project plan to weak? Is there no clear purpose of the project? Has the customer or Beta Software misunderstood the project goals? Do we have communication problems? Do we spend too much time on the project? Are we stressing our customer? Are we missing important phases in the project? These are just a couple of question, and as one thinks, many more questions could be stated. By comparing constructs and understanding why they have strong connections or no connections at al, this information will be valuable while implementing a toolbox. Questions should not only be discussed and documented and used, solution suggestions should also be considered. Looking on different agile methodologies could do this. Once more, questions to discuss and solve this could be asked, such as; does daily scrum meetings reduce the intensity? Could a sprint backlog make the process picture more clear and strong?

By using the results from the data collection, Beta Software will be able to map up both customer types and project characteristics. Moving from the old software process mindset towards this solution is a must if Beta Software wishes to start changing.

5.2 Toolbox implementation

When this case study started, the organization desiderated substratum for implementing a toolbox. At the same time Beta Software described a problem in making profit of an identified software process. This section will present a suggestion of how to start implementing this toolbox in combination with the existing software process. As the organization has put heavy focus on refining their software process, this study has led in the other direction. By analyzing the organizations customer characteristics together with existing communication channels, focus have moved from a mindset were a software process is important to the new way of seeing the organization in relation to their various stakeholders. Putting the customers in focus gives a new solution seen from another angel to the problem described in the problem area.

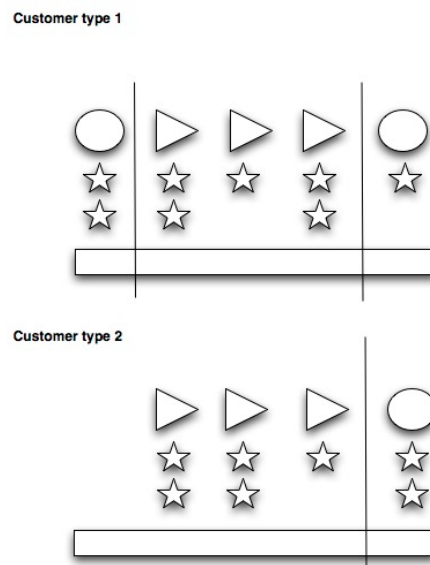


Figure 11. Toolbox implementation suggestion

The above figure still looks like some modification of a software process, but there are differences. When thinking of the toolbox, it is supposed to identify unique project phases and activities for typical customer project types.

Designing the toolbox according to customer types come with several advantages. From one perspective the project managers can fill the toolbox with customized content for typical customers. The content will diversify and might contain e.g. document templates, suggested tools usage and important activities in one view, as another view contains meeting agendas and received feedback. The toolbox view might still look like a software process, but one thing differs. Instead of following a software process with planned activities for each phase, this solution focuses on activities, tasks and

tools that are used during a certain phase. As explained earlier, the different customer projects are unique and it is very difficult to structure a software process that suits all projects. By doing it this way, each unique project has its own defined phases, with personalized content. The defined software process works as a skeleton while setting up the views in a toolbox for different projects.

By implementing the toolbox this way enables project managers to support each other, share experiences, knowledge and data. It is recommended that the project managers meet once in a while, discussing their projects and their personal toolbox content. Agile best practices suitable for different customers can also be stored in the toolbox.

It is also able to use the toolbox concept from a sales perspective. Presenting completed successful projects for new customers with similar characteristics is more reliable than presenting a software process map. Having a discussion with new customer where they may characterize themselves enables collaboration, integration and trust. This concept gives Beta Software the opportunity to package their services in an impressive way.

During the collaboration it was a challenge to convince the organization that change and plans of building a toolbox is connected with an open-minded mindset. Thinking in old courses will not result in productive and innovative solutions. The organization was skeptical to the presented solution. Describing the functions of the toolbox together with the pros and opportunities convinced Beta Software. As they started to understand the whole picture of the collected data of the whole study, they became more positive and positive to the suggestion. By presenting this toolbox concept solution, the organization decided to give it a try, and plans were made to start building the model on the internal share point [12] network. The feeling of being stuck in an old, never updated and nonfunctional software process was as good as gone!

5.3 Moving from ad hoc to agile development

As Nerur [1] describes “The change itself has many perspectives such as communication, documentation, customer relations etc” and “Organizations must carefully assess their readiness before trending the path of agility”, this section will look at some of the perspectives and collaborate around the organizations readiness of trending the path of agility.

As this study comes to an end, its important that Beta Software asks the same question that Nerur does. Are we ready for transformation? It is of highest importance that Beta Software understands the purpose of this case study, together with the execution of it and the founded results from the data collection sources. By understanding how the different results together contribute with material that can be further processed and used while change is upon the organization.

By walking through and educating Beta Software with agile knowledge, it is their task to dig deeper in it, to understand when to use certain principles and why and how they can be used for different customer profiles.

As we have looked on Beta Software’s customer types, analyzed the communication channel, described the desired toolbox and walkthrough agile methodologies hopefully beta industries will realize the advantages of agile development. This study has helped them visualizing their situation in a way that weren’t thought of before. By active participation, it has been discovered that the organization has great possibilities of being an agile organization. Reflecting back to the base for this study, article written by Nerur [1]. This study have helped the organization in matters of “Organizations must carefully asses their readiness before trending the path of agility”. This study have helped them understanding that they have the preconditions to work agile, the agile mindset and the customer characteristics are also. The change itself has many perspectives such as communication, documentation, customer relations etc.

Table 1. Table over agile attributes with explanations

| Attribute | Agile |
|--|--|
| 1. Fundamental assumptions | High-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change [1:75]. |
| 2. Control | People centric [1:75]. |
| 3. Management style | Leadership and collaboration [1:75]. |
| 4. Role assignment | Self organizing teams – encourages role interchangeability [1:75]. |
| 5. Communication | Informal [1:75]. |
| 6. Customers role | Critical [1:75]. |
| 7. Project cycle | Guided by product features [1:75]. |
| 8. Desired organizational form/structure | Organic (flexible and participative encouraging cooperation social action) [1:75]. |

Table 2. Table over Beta Software readiness and described challenges

| Readiness | Challenge |
|------------------|---|
| 1. Ready | Understanding how to accomplish this. |
| 2. Ready | Leaving old waterfall software process visualization. |
| 3. Ready | How to interact as a collaborative project manager / leader. |
| 4. Ready | Seeing each individual's strong side. |
| 5. Ready | Understanding what communication channel to use due various situations. |
| 6. Ready | Understanding why customer's role is critical. |
| 7. Ready | Seeing the importance of product features. |
| 8. Ready | Toolbox, project managers meetings, sharing knowledge. |

As this article extends Nerurs' [1] describing challenges, the top table lists some key words for software development followed by an explanation how to manage them in an agile way. The second table describes that Beta Software are ready to follow these described directions, followed by a described challenge for each attribute.

By providing concrete examples in the list below, Beta Software will be able to handle the previous described challenges while moving to agile development.

1. By using small teams with good customer interaction, valuable feedback can be assessed and successful projects can be delivered. As the organization is a small company, they will be able to use small strong teams interacting with their different customers.
2. Beta Software must move their focus from a software process visualization and understand how communication, peoples skills and customer profiles will contribute more to the organization than trying to visualize some type of software process that lacks some important steps in software process development. As the organization has material from this collaboration, supporting both communication and customer profiling, they have a great advantage while focusing on and extending the mentioned tasks.
3. By following the principles of scrum and arranging daily scrum meetings, the project managers will collaborate with their developers and being able to give both directions and support for the various projects a project manager may be involved in. If the organization is unsure how to enable this, looking back on the related research section of this paper may help them doing so.
4. The informal question time at Beta Software has revealed that the employees possess different strengths and specialties. Project teams should include persons with some common skilled needed for the project, together with some unique strengths. This follows some guidelines, described by Cockburn, "Agile processes are design to capitalize on each individual and each team's unique strengths" [16:132]. As this study reveals, the project managers in Beta Software are the main characters in the projects. As they interact with both customers and developers it is important that they understand each unique strength and specialty of both their developers and their customers.
5. As this study has analyzed the communication channels, it is important for the project managers to understand when to use different communication tools. It is also important to

work on making some channels stronger than the results reveal. Realizing the advantages of communication compared to documentation is an important task for the project managers' path of agility. It is of high importance to have a discussion around the results from the communication channels analysis done during this study. Doing so will help the organization understand why they should communicate with a certain type of customer in a certain way.

6. By studying the results from the second interview that presents customer characteristics, the project managers' will understand each customer's unique characteristics and the impact they might have on the project. As a new project starts, together with the customer a discussion should take place, where the customer is able to put themselves in a category, described by some attributes taken from the constructs in the second interview. By doing this, project managers can show successful projects with the same properties and give examples how they were managed.
7. In chapter 4.1, the project manager's interpretation of the software process was presented. The process map looks very "waterfall" and traditional, and must be redone, in order to enable an agile mindset and understand the differences between the old and new one. The project cycles focus must be taken from process to features. Below an example will be given.
8. By implementing the previously described toolbox, the project managers' will have great support for being an organic and flexible organization. As a project runs into problems, discussion with other project managers where knowledge, management and guidelines are considered, collaboration and social action is enabled.

Beta Software has great preconditions for running agile projects, the issue is more realizing how to accomplish, adapt and think agile. As this study has been done, both readers of this document and the organization will understand how the presented data results enable the opportunities of being an agile organization.

5.4 Contributions and achievements

This collaboration has helped Beta Software with their three main problem areas; the gap between theoretical models and actual ways of performing work, urge of a toolbox and the transition from non-functional ad hoc software development to agile software development.

The organization has been provided with supportive material for all three mentioned problem areas. Beta Software has showed a mature attitude towards criticisms, as well as improvement suggestions. Instead of failing in trying to solve these problems in a certain way, they have started to think on their own and been able to see the problems from other perspectives.

The way that Beta Software has received material has been satisfying from a researcher's point of view. The organization has both understood the results from various data collection sources, as well as the opportunities and abilities that can be done by further elaboration on presented results. Beta Software has material to start building on a toolbox. This material may also work as support for the project managers in their daily work. The organization has also understood how to choose agile principles when collaborating with different types of customers. This could be accomplished by understanding results from the different interviews. If the organization can manage to use the toolbox frequently, and simultaneously choose agile principles while dealing with new and old customers, they will have a great starting point in sense of transition. The described gap will be reduced by these two mentioned accomplishments.

6. Conclusion

Change in an organization is a difficult activity. Throughout this paper we have read about Beta Software and their case. As the study started, the organization experienced a gap between theoretical models and actual ways of structuring and performing every day work. This was followed by lack of knowledge in agile development and a desire of support material for implementing a toolbox. This study has provided Beta Software with material enabling change progress in all three mentioned areas. As the organization understands their different customer profiles and communication channels they are now able to start implementing a toolbox. This toolbox will contain different views for the customer profiles and together with that, typical process phases will be listed were belonging material will be categorized. As the company now knows their typical customer profiles, this study suggests suitable agile practices for certain customer profiles. E.g. it might be more beneficial to run a certain agile practice on a distributed and technically skilled customer compared to a co located and technically unskilled customer, as the discussion chapter [5.1.1, 5.1.2] recommends. As the discussion points out two different examples, it is Beta Software's duty to make further elaborations in identifying suitable work practices for different customer types.

As this research is done for Beta Software and the results are collected for their unique cases, it might be difficult for organizations with similar problems to fully understand the strength and results from this study. Therefore other companies should consider if their organization is functional or if improvements are either crucial or worth investigation time. As other organizations might find issues and problems in their development chain, it could be worth trying new methods and looking on these problems from a different point of view. This study is just another way of researching in a problem area and an attempt of providing material for a better functioning organization. As people's mindset, technique and development methods change it is important to adapt. In this study some agile methodologies are briefly presented and described in relation to the organizations data results. There are many agile practices while developing software. Organizations with similar problems to Beta Software must understand the time consuming process of finding usable material and research methods before heading into an eventual transition.

Concerning the contribution to humans interesting in agile software development has this study also provided another way of using these methodologies. By focusing on customers, it has been revealed that different agile methods are more or less suitable in different cases. There is probably not one ultimate solution, sometimes one must combine ways of working and interpreting from one ore more agile methodologies. Running agile projects is not a guarantee for a successful project and therefore organizations should ask themselves why they want to work agile, how it can benefit them and how it will affect their customer relations. Before moving from ad hoc to agile software development it is most important to have a clear goal, and an understanding of the expectations while doing so.

RERERENCES

- [1] Nerur B, Mahapatra R, Mangalaraj G. Challenges of migration to agile methodologies – Organizations must carefully assess their readiness before treading the path of agility. *Communication of the ACM*. May 2005.
- [2] Rising L, Janoff N.S. The scrum software development process for small teams. *IEEE Software*. July/august 2000.
- [3] Schwaber K. Scrum development process. OOPSLA Business Object Design and Implementation Workshop.
- [4] Abrahamsson P, Salo O, Ronkainen J, Warsta J. Agile software development methods -Review and analysis. VTT Publications 478. 2002.
- [5] What is extreme programming? [Accessed: May 21 2009]. Available from: <http://www.xprogramming.com/xpmag/whatisxp.htm#whole>
- [6] Tatum R. Applying lean thinking principles to software development. University of Oregon. June 2005.
- [7] Raman S. Lean software development: Is it feasible?. *IEEE*. 1998.
- [8] DSDM Consortium – Enabling business agility. [Accessed: May 21 2009]. Available from: <http://www.dsdm.org/>
- [9] Voigt B.J.J. Dynamic system development method. University of Zurich. January 2004.
- [10] Qualitative research in information systems. [Accessed: May 21 2009]. Available from: <http://www.qual.auckland.ac.nz/>
- [11] Walsham G. Interpretive case studies in IS research: nature and method. *Operational research society*. 1995.
- [12] Microsoft SharePoint Server – Connecting People, Process and Information. [Accessed: May 26 2009]. Available from: <http://sharepoint.microsoft.com/Pages/Default.aspx>
- [13] Maruping L.M, Agarwal R. Managing team interpersonal process through technology: A task – Technology Fit perspective. *Journal of Applied Psychology*. 2004.
- [14] Tan F.B, Hunter M.G. The repertory grid technique: A method for the study of cognition in information systems. *MIS Quarterly*. March. 2002.
- [15] REP IV. [Accessed: May 21 2009]. Available from: <http://repgrid.com/>
- [16] Cockburn A, Highsmith J. Agile software development: The people factor. *Computer*. November 2001.
- [17] IEE Xplore: Guest Home Page. [Accessed: May 26 2009]. Available from: <http://ieeexplore.ieee.org/Xplore/guesthome.jsp>
- [18] The ACM Portal. [Accessed: May 26 2009]. Available from: <http://portal.acm.org/portal.cfm>
- [19] Google scholar. [Accessed: May 26 2009]. Available from: <http://scholar.google.se/>
- [20] Manifesto for Agile Software Development. [Accessed: May 26 2009] Available from: <http://www.agilemanifesto.org>
- [21] Wiki – Wikipedia, the free encyclopedia. [Accessed: July 21 2009]. Available from: <http://en.wikipedia.org/wiki/Wiki>