



WE ARE ALL US

Case study on implementing agile methods in an organization

Helinä Pylvänäinen

**Master's thesis
May 2009**

School of Business Administration



JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

Author(s) PYLVÄNÄINEN, Helinä	Type of Publication Master's thesis	
	Pages 108	Language English
	Confidential <input type="checkbox"/> Until _____	
Title We are all us Case study on implementing agile methods in an organization		
Degree Programme Master's Degree Programme in Entrepreneurship and Business Competence		
Tutor(s) MÄNTYSAARI, Anne NUKARI, Jussi		
Assigned by Corporation xxxxxxxxxx, Business Unit xxxxx		
Abstract <p>In the 21st century, agile software development methods have been exploited by numerous enterprises. Several studies argue agile methods and lean management to have significant impact on productivity and efficiency. Enticed by these promises even more and more software companies and development teams decide to abandon old waterfall approach and implement the new way of developing.</p> <p>Business Unit xxx's business is based on repeatable solutions aka products. Product development is carried out in release projects, which have several deliveries annually. Development project members reside in several countries. Defined processes, methods, and practices are followed and used in both project management and product development.</p> <p>However, using traditional processes and methods in project management and product development turned out to be somewhat inadequate, cumbersome and time-consuming. Since the time-to-market was considerably long and the productivity and efficiency of the projects were claimed to be inadequate due to inflexibility in project management and development processes, some measures had to be taken. Agile methods and practices were therefore implemented to overcome the current problems as well as make development more lean and efficient.</p> <p>The purpose of this thesis work is to describe the transition and to reflect on the consequences caused to the entire organisation when roll-outing agile methods in product development. Project stakeholders were interviewed and some improvement ideas collected.</p> <p>The major results of the study emphasise the importance of implementation to create a new agile culture to the entire organisation to fully exploit agile methods. This requires changes in management, reporting, measuring, and collaboration. Evaluating, reflecting, and adjusting the used methods is a never ending endeavour.</p>		
Keywords Projects, management, change, software, development		
Miscellaneous Appendix 4 removed from electronic format. Name of the company changed in the electronic format.		

Tekijä(t) PYLVÄNÄINEN, Helinä	Julkaisun laji Opinnäytetyö, ylempi-amk	
	Sivumäärä 108	Julkaisun kieli englanti
	Luottamuksellisuus <input type="checkbox"/> Salainen _____saakka	
Työn nimi We are all us Case study on implementing agile methods in an organization		
Koulutusohjelma Yrittäjyys ja liiketoimintaosaaminen		
Työn ohjaaja(t) MÄNTYSAARI, Anne NUKARI, Jussi		
Toimeksiantaja(t) Yritys xxxxxxxxxx, Liiketoimintayksikkö xxxxx		
<p>Tiivistelmä</p> <p>Perinteiset projektinhallinta- ja tuotekehitysprosessit ja -menetelmät korostavat huolellista ennakkosuunnittelua ja seurantaa sekä työskentelyn vaiheittaista etenemistä siten, että edeltäviin vaiheisiin ei yleensä palata takaisin. 2000-luvulla ovat suosiotaan kasvattaneet ns. ketterät menetelmät, joita tutkimusten mukaan käytetään laajasti tuotekehitysprojekteissa. Ketterien menetelmien odotetut hyödyt ovat tuottavuus ja tehokkuus koska mm. projektinhallinta, päätösprosessit, organisaation roolit ja dokumentointi on vähennetty minimiin. Tiimit toimivat itsenäisellä tavalla jatkuvasti omaa toimintaansa arvioiden ja kehittään. Projektipäällikön tehtäväksi jää lähinnä tukeminen ja tiimin työtä haittaavien esteiden poistaminen.</p> <p>Liiketoimintayksikkö xxxxx:n tuotekehityksessä on käytetty perinteisiä prosesseja ja menetelmiä. Ongelmat tuottavuudessa, tehokkuudessa ja laadussa herättivät kiinnostuksen uusiin ketterämpiin menetelmiin tavoitteena nopeampi uusien tuotteiden tai tuotepiirteiden kehittäminen ja nopeampi reagointi markkinoissa tapahtuviin muutoksiin. Ketterien menetelmien käyttö aloitettiin kolmella pilottiprojektilla. Saatujen tulosten rohkaisemana menetelmiä alettiin käyttää kaikissa tuotekehitystiimeissä. Käyttöönottovaiheessa huomattiin merkittäviä vaikutuksia sidosryhmiin. Muutostarpeet yhteistyössä ja muissa prosesseissa nousivat keskusteluissa tärkeälle sijalle. Tämä tutkimus on tehty selvittämään millaisia muutoksia sidosryhmien toimintaan ketterien menetelmien käyttöönotto tuo ja ulottuvatko vaikutukset myös johtamiseen.</p> <p>Tutkimus on tehty tutustumalla aiempiin tutkimuksiin ja muuhun kirjallisuuteen, tarkkailemalla ja osallistumalla käyttöönottovaiheeseen, henkilökunnan tiedottamiseen ja koulutuksiin ja sekä haastatteleamalla eri rooleissa toimivia henkilöitä käyttöönotosta ja sen vaikutuksista.</p> <p>Tulokset vahvistavat tutkimusoletuksen, että ketterien menetelmien käyttöönotto tuotekehityksessä heijastuu muuhun organisaatioon, jonka toimintatavoissa tulee tarvetta muutokseen. Lähijohtamisessa tapahtuu myös merkittäviä muutoksia, mutta liiketoiminnan johtamisen muuttumiseen ei havaittu selkeää yhteyttä.</p>		
Avainsanat (asiasanat) Projektit, projektinhallinta, tuotekehitys, johtaminen		
Muut tiedot Liite 4 poistettu sähköisestä versiosta. Yrityksen nimi poistettu sähköisestä versiosta.		

CONTENTS

1 INTRODUCTION.....	3
1.1 Background.....	3
1.2 Initial state.....	4
1.3 Research question and research method	6
1.4 Presenting the case	11
1.5 Structure of this report	12
2 PRODUCT DEVELOPMENT.....	13
2.1 Product business	13
2.2 Agile methodologies.....	16
2.3 Project management.....	18
2.4 Software development	23
2.5 Agile management.....	27
2.6 Organizational changes	31
2.7 Agile tools	34
2.8 Remarks of the literature review	34
3 AGILE METHODS AS A TOOL FOR EFFICIENCY	36
3.1 Expectations	36
3.2 Organising product development	38
3.3 Product lifecycle.....	40
3.4 Roll-out project.....	41
3.5 Means: practices and tools	45
4 AGILE METHODS IN USE	47
4.1 Achievements	48
4.2 Effects on the organization	51
4.3 Effects on the management	55
4.4 Findings from the study.....	59
5 CONCLUSIONS	61
REFERENCES.....	67
APPENDICES	72
Appendix 1: Definitions	72
Appendix 2: Scrum checklist by Henrik Kniberg	74
Appendix 3: Case study: Cover letter and questions	75
Appendix 4: Case study: Interview answers	78

FIGURES

FIGURE 1. The structure of the study	8
FIGURE 2. Research project iterations	10
FIGURE 3. The influence of stakeholders and cost of changes.	21
FIGURE 4. Overview of Scrum agile software development	22
FIGURE 5. Product Line Teams: Release pattern.....	39

1 INTRODUCTION

This thesis is a case study which analyses a transition of one organisation from traditional project management and software development processes to more agile ones in Business Unit xxxxx which is a sub-business unit in Corporation xxxxxxxxxxxx. Such an extensive shift needs commitment and collaboration of the entire company to succeed, albeit can also lead to “We are all us” community spirit. The introduction presents the background for the transition, the initial state of processes and the case. Additionally, the research questions and methods are described. The study consists of presentation of traditional and agile project management and software development and of an empiric study on the case. This report is written in an iterative manner, so what is initiated in introduction will be more widely described in the theory section and even more deeply discussed in the empiric section.

1.1 Background

The popularity of using agile methods instead of traditional waterfall approach in software development has increased in the 21st century (Cohn & Ford 2003, 74). Most wanted consequences for using agile methods are productivity, efficiency, lean management and quality of the software. However, agile methods are easy to misunderstand and hard to get right (Heusser 2006, 1) problems can occur in technical, cultural, or organizational issues (West 2009b) and possible side effects can consist of lack of structure creating chaos (Highsmith 2004, 17), using change as an excuse for project problems (West 2009a, 16) etc.

In Business Unit xxxxx, the business is based on repeatable solutions also called as products. The core products are sold to several customers with additional features and only few customer specific features are included. Customers’ requirements are collected and evaluated by customer and sales unit to create a product roadmap. New features and changes are ordered from product development on yearly basis. Product development is carried out in

release projects, which have several deliveries annually. Development project members reside in several countries. Defined processes, methods, and practices are followed and used in both project management and product development. There is no common formal software development model in place but a waterfall oriented approach is generally used.

Since the time-to-market is considerably long and the productivity and efficiency of the projects are claimed to be inadequate due to inflexibility in project management and development processes, some measures have to be taken. In one sub-business unit, agile methods have been successfully used for several years, resulting in quick time-to-market and high product quality. An initiative to analyse and pilot these methods in order to see their suitability in product development was raised.

While an organisation takes agile software development methods in use it has consequences to the entire organisation and failing to introduce the change to every stakeholder will have negative impact on the results (Cohn & Ford 2003, 74). If the preferred improvements in the product development are expected, the product portfolio management, project administration, and delivery model have to change, too. Introducing change to an organisation is difficult and confronts resistance (Cockburn 2007, 319) but to succeed in future business environment the company has to develop and change continuously (Kettunen 2008, 9).

As Kainulainen (2008, 84) states in his study, no significant improvement will be gained of agile methods if they are not properly implemented and if the management is not committed to the new way of working. Quite the contrary, the productivity may decrease and the time-to-market increase if the old processes and practices are not abandoned and the new ones properly implemented in agile development teams (Kainulainen 2008, 85).

1.2 Initial state

Using a traditional Project Management process has been the company policy for several years. The process was developed particularly for the company

and wide range of document templates, methods and business models have been developed and adopted to the process. In addition, project managers are encouraged to certificate with IPMA certification, which also supports traditional way of managing projects.

Software development process in Business Unit xxxxx has basically been following the traditional “waterfall” method with requirements, analysis and design, implementation, and testing, verifying, and validating phases. In these phases appropriate tasks are carried out, as well as, appropriate documents are produced as input and output of each phase. The development process is basically based on CMMI model (= Capability Maturity Model Integration) requirements. Since the development projects and the process itself shall regularly be assessed to measure the state of operations for improving the process, project, and product quality (Ambler & Kroll 2007b) frequent reviews and audits have been conducted.

However, using traditional processes and methods in project management and product development has turned out to be somewhat inadequate, cumbersome, and time-consuming. It also creates frustration (Boehm 2007, 74). For example, a significant amount of development project’s time is spent on project administration for writing mandatory documents and following through recommended methods. In addition, communication and information sharing is troublesome, since development is carried out in distributed teams in several countries and project organization consists of several groups and levels. Managing the incompleteness of communication is therefore a special focus area in agile software development (Cockburn 2007, 1).

Nevertheless, the traditional methods have advantages, too. Capable and certified project managers are available and the project workload and schedule are rather well estimated from the beginning of the project. In addition, definite dates can be given for delivery as well as resource allocation of specific resources, such as testers, who are only needed for a short period of time.

1.3 Research question and research method

I started to plan my Master's thesis in the spring 2008, approximately at the same time as some agile practises were injected into pilot projects. First, I thought the research question would be studying which agile practises could be used in Business Unit xxxxx product development and how they should be implemented to exploit them best.

In Business Unit xxxxx, a global delivery model is used meaning that the software development teams are not compact onsite teams. Instead it is distributed to multiple teams in several countries within two continents and three time zones. This creates pressure on the methods, communication and management more than traditional onsite development teams (Moore & Barnett 2004, 5). The agile methods are originally meant for small onsite teams (Kusssmaul 2004, 126) although there are examples of successful exploitation of them in complex multi-team environment (Sutherland, Viktorov & Blount 2006, 3).

After reading substantial amount of theory I understood that there already are reasonable amount of research done about implementing agile methodologies and practises in different kinds of software development teams. Dybå and Dingsoyr (2008, 1) state that there are total of 1996 theoretical studies about agile. The amount of empiric studies is significantly fewer though, only 36 studies had been made until 2006. As conclusion, it did not seem probable to get major results on that line of study.

While the roll-out of the agile methods spread through the entire development organization in the end of 2008, more and more people were affected by these new ways of working. For example, the product development projects have stakeholders from several teams and different functional interest groups. Increasing amount of questions was raised and many of them were concerned inquiries of how the rest of the organization should change and adapt: which processes need fine-tuning, what management procedures should be cut down, and how the actual change process should have been carried out.

Finally, the research framework and theme started to form of current interest: how does agile product development affect other functional operations such as product management, customer support, delivery and deployment, testers, sales, and line management. In addition, does this impact even reach-out to business processes: annual business planning or product road mapping. Management in particular is said to resist the adoption of agile methodologies since they seem to bring quite dramatic changes to the way of working and thinking (Augustine & Woodcock 2003, 3), does this happen in our organizations and can it be seen somehow.

The final research questions are:

1. How does adopting agile methodology into product development affect different stakeholders and the entire organisation?
2. What difficulties were identified during the implementation of agile practises in product development?
3. How should the change be managed and how former management practises will change?

I do not have a predefined hypothesis except the assumption that the implementation of agile practices in the product development will have some effect on the entire organisation. This hypothesis is based on the preliminary observation during the early days of adoption.

Figure 1 shows the structure of the study. The theory part consists of background information of agile methodologies, project management, software development, organizations, management and agile tools. The empiric part presents the exploitation of agile methods in Business Unit xxxxx; the expectations and the actions. Additionally, the consequences of transition to organization and management are described.

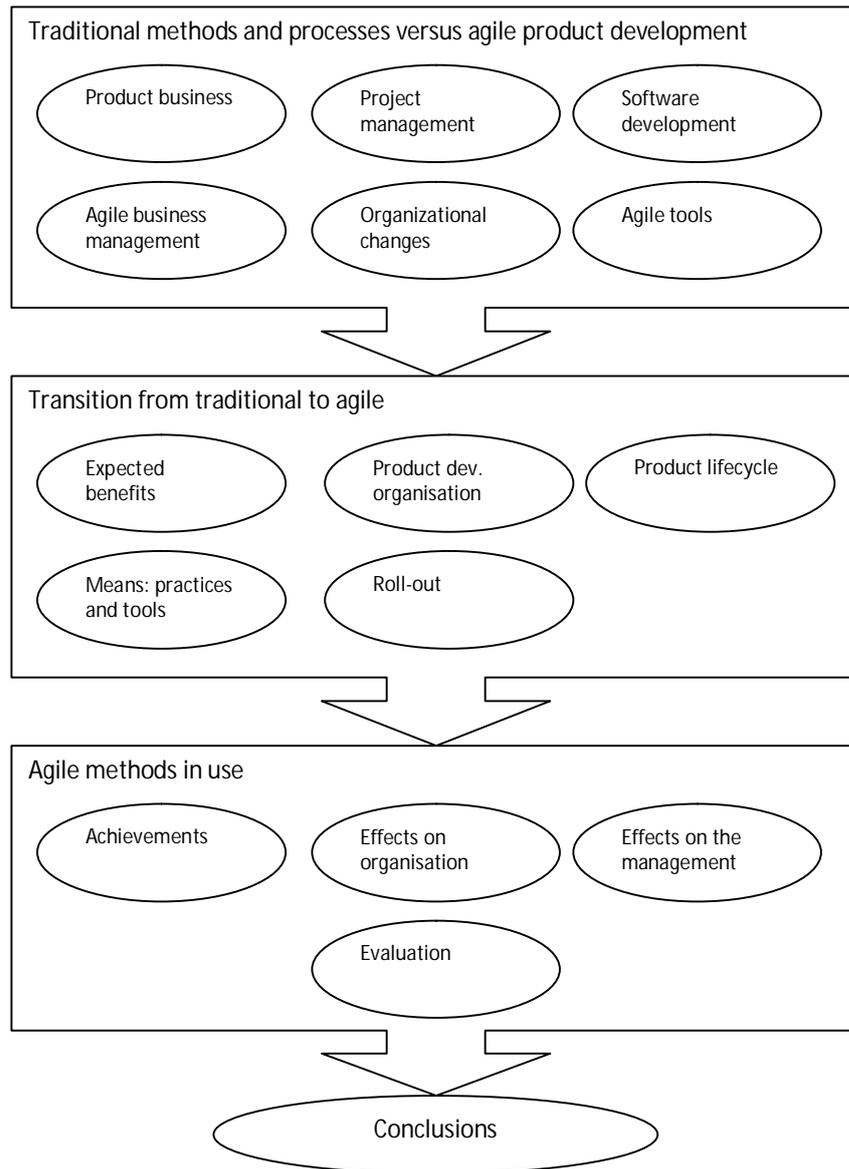


FIGURE 1. The structure of the study

In this study, a qualitative approach has been utilised. The theory is based on reading of agile studies, white papers, and documented experience of agile exploitation (mainly from internet sources). Several of the books and articles are published by practitioners and consultants of some agile methodology and not by academic researchers. This creates therefore the need to review them critically and in some cases to compare with each other. Nevertheless, significant amount of literature is available and it is only possible to review and refer to some of them.

The empiric study is done by attending to information meetings and trainings, studying presentations spread to the organisation, and interviewing several

stakeholders of diverse roles and responsibilities in product development business on the case on implementing agile methodologies in product development during years 2008 and 2009. The documented interviews have been collected during spring 2009 while most of the subjects had approximately six months of experience of agile. The subjects to the interview were chosen based on their role in the organization (see appendix 3) to represent main functionalities, all four operating countries, and different management levels.

Since response rate of surveys can be low, the subjects may not take the questions seriously, or they do not understand the questions properly (Hirsjärvi, Remes & Sajavaara 2000, 182), I chose to use interviews instead of survey. In addition, hereby the subjects of the interviews can be more carefully chosen to represent diverse functionalities and processes in the organization.

Methodologically the approach used in the interviews represents a qualitative interview investigation through seven stages: thematizing, designing, interviewing, transcribing, analyzing, verifying, and reporting (Kvale 1996, 88). The interview questions are partly structured and partly open, both based on the stated research themes and questions. Open questions create an opportunity for the subject to raise issues they find important.

Thematizing and designing of the interviews are based on the background information of the company and its business and the review of theory and previous studies in the area. The interviews are exploratory interviews within given themes and they have only little structure. The purpose of the study is firstly to identify all stakeholders that are affected when software development teams utilise agile methods, secondly to identify possible problematic areas and questions which have been raised during the implementation and finally to find out how the roles and processes of management will be changed. For this reason, the interviews are not merely for testing a pre-designed hypothesis but also to gather open information and opinions from diverse stakeholders.

Interview questions were send to the subjects beforehand and the interviews were carried out mainly by telephone, some in face-to-face contact, though.

Transcribing the interviews to text is based on the notes of both the subject and the interviewer. Meaning condensation and categorisation were used when analysing of the interview text. Categories, such as expectations, roll-out, and achievements, were created based on the themes of the study. The interviewer restrained herself from interpreting the answers. Instead the unclear answers were verified from the subjects. A synthesis of the interviews, observations of the case, and published material for the transition are used to present the empiric results.

To verify the generalizability, reliability, and validity of the findings and result, they were compared with the theoretical basis. However, the analysis of a case study like this cannot be generalized and the results transferred into another organization as such; only some trends and guidelines can be drawn. This is due to the characteristics of a qualitative study, it cannot be repeated and exactly the same results collected again. The case is unique in measures of the implementation phase and organizational roles. Interviewing the same subjects after one year would (hopefully) give different results. The conclusions chapter in this report presents the final remarks, the main results, and suggestions for further study.

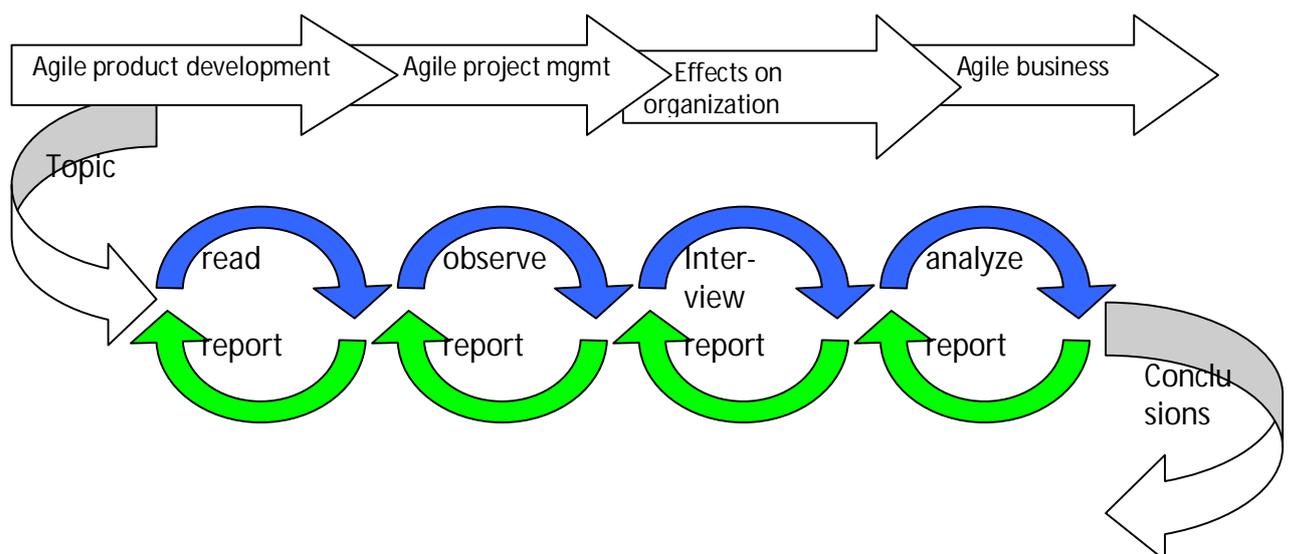


FIGURE 2. Research project iterations

In Figure 2, the research project is illustrated as an iterative process. The project started with overall definition of the topic which was adjusted after

each iteration. First iteration: the reading, analysing and reporting of theory of agile product development. Second iteration: observing pilot projects. Third iteration: making interviews of different stakeholders. Fourth iteration is for analyzing the interviews, making a synthesis of the case, and comparing the results with the literature. Finally the project ends with the conclusions.

1.4 Presenting the case

In one sub-unit of Business Unit xxxxx, agile methods have been used since 2004 with good results in an onsite development team. In the spring 2008, three experimental pilot projects were started in other development teams to see whether agile processes and methods could help in making the development projects more lean and efficient in a distributed team organisation. In addition, the aim was to find out whether product development in overall could be carried out in an agile manner.

Since the piloting proceeded well with promising results, all development teams started to exploit some agile methods and practices by the end of 2008. One of the prerequisites of using agile methods was however that they should not affect the rigor policies of delivery dates and controlled management on integrated projects. In addition, the predictability of scope is also seen as a valuable asset since product development outcome, such as product changes or a new feature, will be delivered and have to be valued by mass market. In comparison to changes in a tailored solution to only one or selected key customers this means fulfilling the expectations of dozens or hundreds of customers at the same time. Thus the scope cannot easily be negotiated with the customers.

Now, one year has passed from the beginning of first experimental projects and approximately six months from the rest of the teams following. The teams have done frequent reflection and improvements according to agile principles while they have tried out new agile practices and methods. While these different agile methods and practices have been injected into development projects, the evaluation and reflection on their effects is important (Cockburn 2007, 323); to find out which practices help to achieve the objectives and

which do not work as expected or do not bring significant advantage to the previous processes or even decline the earlier performance.

Inevitably the utilization of new processes, methods and practices has influence and dependencies outside the development team. Other stakeholders, such as product management, delivery and deployment, customer support and sales, must adapt their ways of working in order to efficiently cooperate and collaborate with the development team. The objectives and achievements as well as the possible clashes between different stakeholders have to be identified and the problems or obstacles analysed in order to remove the impediments for efficient and productive work.

Finally, the problems encountered by different stakeholders and possible recommendations of solutions for overcoming them have to be collected and handed over to management review if they cannot be solved within the development team and appropriate stakeholder. The management has the definite responsibility to evaluate and decide whether agile practises should be used and in which extent in Business Unit xxxxx, to reduce time-to-market, to increase efficiency and productivity, and to raise the quality of the products.

1.5 Structure of this report

First, I will define and describe the framework for this study. Software development in product business has its own unique characteristics since dozens or hundreds of customers use the same product. Agile project management differs from traditional project management and agile software development from traditional waterfall development model. Management processes and practises are an important skeleton for the company to ensure sustainable success in the business.

Beyond management, product development and the overall organization create complex chains or a network which is needed to manage the product lifecycle from cradle to grave. Global delivery model includes several distributed teams developing products which are intertwined with each other and delivered to different markets. Different agile methodologies introduce

various collections of agile practices and methods but they have some common underlying principles and characteristics. Some of them are shortly listed and references for more detailed descriptions provided.

Second, I will introduce the results from the interviews and observations of the transition. The reasoning behind implementation of agile methods as tool to increase productivity, efficiency, and quality is clarified. Possible remarks of the roll-out process are raised. The changes in the processes, tools, productivity, and organisation are described as they are identified by the interviews. The old and the new processes clashed creating confusion, raising questions and challenging the management to take responsibility of the change. Business and product management has to adapt to these changes to ensure firm support to new endeavours as well as business profitability in changing environment.

Finally, I will conclude by pointing out some findings from the interviews and the study, listing problem areas and issues that need solving, evaluating the most important lessons learned, and giving some recommendations what should be considered in Business Unit xxxxx and in future agile implementation projects.

2 PRODUCT DEVELOPMENT

2.1 Product business

Product business also called as Repeatable solution business (RSB) is based on software products sold to several customers. Product management is responsible for planning and managing the product portfolio as well as identifying the requirements and planning an individual product's roadmap and releases. The product lifecycle of one product can be 10 to 15 years from the first release to the end of maintenance. Product development is based on requirements from various sources: collected from the customers, law demands, and technological changes. Product development is responsible of implementing the products and changes in product development projects.

Business Unit xxxxx has a portfolio of Welfare and Education repeatable solutions that is products, mainly used for managing social benefits such as, maintenance allowances, transport subsidies, and children day care applications placing, and invoicing. Products are grouped into several product lines such as Family care, Elderly care, and Education. A primary benefit of a product line is that new products are built by utilising the common core assets, rather than by creating from scratch (Kussmaul 2004, 127-128).

A product can have many meanings but here it is defined as a software application of which a licence is sold or an ASP (application service provider) service contract is made with the customer. The concept repeatable solution refers to the product as a packaged solution, with configuration parameters, which can be deployed by several hundred customers with no or minor customer specific adaptations.

Customers are mainly municipalities in Finland, Sweden, and Norway. In addition to the licence, customers pay maintenance fee to get support and frequent releases of the application. Furthermore, some more services are provided for the customer, such as delivery and deployment project services, trainings, user instructions, technical support for maintaining customer environments etc., to create a whole product (Moore 2005, 104).

Product portfolio management represents the customer in the development organisation (Braithwaite-Lee 2002) and is responsible of creating a competitive product portfolio for the intended target market. This includes planning and managing each product for entire duration of their lifecycle. A product's lifecycle consists of several phases from the initial idea to the end of maintenance and support. Product lifecycle management is a business activity which has several levels: strategic level with long term road mapping to ensure competitiveness; tactic level focused on improving processes; and an operational level focused on efficiency by improving individual activities (Stark 2006, 16).

Product live cycle can be illustrated as a stretched S-curve having three phases from the viewpoint of the markets: introduction and early adoption, acceptance and growth of the market, and maturity with market saturation (Meredith & Shafer 2002, 29). Stark (2006, 17) lists the product lifecycle phases from the viewpoint of product management as follows: imagine, define, release, support & service, and retire. Most of the products in Business Unit xxxxx are currently in support and service phase and spread to the mainstream markets; basic architecture and design is ready, additional features and error corrections are delivered in frequent releases. Of course there are products that are close to Retirement phase and some that are in Imagine or Define phases. This is typical to a healthy product portfolio.

The basic objectives of product management are to create a product that meets today's customer requirements and delivers customer value (Highsmith 2004, 6). The requirements and the perception of value are diverse: customers focus mainly on usability and cost-effectiveness of the product (Anthes 2005, 27-36), product management has to pursue business objectives, too (Stark 2006, 35). Product roadmaps are planned for three years and they are annually updated. They consist of major functional enhancements and technology changes. In addition to that, there are product wish lists and error lists containing minor issues from customers or from internal sources. Agile methodologies do not expect that all requirements are collected up-front (Stepanek 2005, 91); the customer feedback after each iteration specifies and directs the next increment (Beck & Andres 2004, xvii).

Product releases are planned according to roadmaps, product wish list and error lists. The release schedule consists of annual major and minor releases which are shipped according to predefined schedule. Value to the business is not immediately realised on product or feature delivery, only some time afterwards. According to agile principles, value has to be delivered early (Abrahamsson, Salo, Ronkainen & Warsta 2002, 12) and the most important features are implemented first (Abrahamsson et al. 2002, 16). This means earning sooner to cover the costs coming later (Beck & Andres 2004, 12). In incremental deliveries the payback time of investment is only half compared to traditional projects (Scumniotales 2009).

Product development is done mainly in Norway for Norwegian markets and in Sweden, Finland, and India for Swedish and Finnish markets. The development teams are distributed to these countries and several of the teams have some members in India for getting development costs lower. Rapid changes in markets and increased emphasis on usability and value (Boehm 2007, 717) put high pressure on project management while managing these teams and the development projects.

2.2 Agile methodologies

Agile is a concept that is used in many contexts. Just to name some, currently agile project management, agile software development, and agile business management are widely discussed in literature and in the internet. Despite the recent interest, agile principles have a history of some decades; already in 1986, Boehm introduced a spiral model where a handful of requirements were planned, designed, and implemented and customer review collected before moving on to next bunch of requirements (Coplien 2008).

Agile project management mainly introduces roles and responsibilities, meeting conventions, requirements managing, and other structural practices and tools to manage projects in a way that does not have excessive or superfluous management - only sufficient to succeed in the project. These practices are based on experience rather than theory (Stepanek 2005, 65).

Agile development methodologies consist of software engineering techniques such as pair programming, continuous integration, or collective ownership of the code, to produce software efficiently and with good quality (Schuh 2005, 48).

Change management has always been important for business to survive in changing situations. The markets, the competitors and the consumer behaviour are said to change more and more rapidly in these days. Agile strategies and business management is therefore said to be vital to survive in future (Doz & Kosonen 2008, 275).

A methodology is a set of conventions that a group of people agrees to follow whereas a method refers to techniques or procedures (Cockburn 2007, 149). These conventions are likely to change over time (Cockburn 2007, 209). Agile practices are the tools of the agile trade: they are errands agile teams regularly do to stay disciplined, foster communication, remain flexible, and deliver software (Schuh, 2005, 4).

Well known and commonly used agile methodologies (West 2009a, 3; Schuh 2005, 2) are: Scrum, Extreme Programming (XP), Test-driven development (TDD), Crystal, Open Unified Process (OpenUP), Lean Software Development, Feature-Driven Development (FDD), Adaptive Software Development (ASD), and Dynamic System Development Method (DSDM). Significant amount of literature, articles and studies have been written to describe them and their application in software development teams. See for example: Abrahamsson et al. (2002), Ambler & Kroll (2007a, b, c) Balduino(2007), Cohn (2005), Gustafsson (2008), Kainulainen (2008), Kalermo & Rissanen (2002), Kniberg (2007), Kroll (2007), Loeser(2006), and Schuh (2005).

The Agile Manifesto underlines the common principles or core values of different agile methodologies' practitioners (The Agile Manifesto 2001):

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Highsmith (2004), who is one of the subscribers of this manifesto, emphasises that although individuals and interactions, working software, customer collaboration, and responding to change are seen more important or even critical this does not mean that the processes, tools, documents, contracts, or plans are unimportant; they are just less important in agile methodologies.

Different agile methodologies have many distinctive features in common. Here is a listing of the properties Cockburn (2008) states to be in common for highly successful projects:

1. short iterations, which are followed by frequent deliveries. In other words, developing a small defined part of the software at a time and delivering that part before continuing with next tasks.
2. frequently reviewing iterations and making reflective improvement based on them
3. close and constant communication within development team, which usually requires a collocated team and shared team room
4. open atmosphere for discussing problems, progress, and reflection and safety to convey personal opinions
5. focusing on the current project and task without unavoidable interruptions. That is, one developer works within one project at a time and can concentrate on the tasks in hand.
6. an easy access to expert resources, for example end-users of the software (or their proxy), is essential for getting accurate feedback and answers to probing questions
7. advanced technical environment with frequent builds, integrations, and testing.

The agile methodologies not only present common principles and features but also a variety of practices and tools to be used in project management and software development. For example, OpenUP presents more roles, disciplines, tasks, artefacts, and step-by-step explanations of the process (Balduino 2007) than Scrum which only has three roles, minimum set of practices, and no fixed process descriptions (Schwaber & Beetle 2002, 7, 100).

2.3 Project management

Traditional project management has been used for few decades; Project Management Institute (PMI) in USA and IPMA in Europe promote project management to businesses and organisations (IPMA). They have established project management standards, provided seminars, educational programs, and certifications to increase the professionalism in project management (PMI). However, the software development projects and especially emergence of agile software development methodologies have created new practises to

project management, too, since traditional project management is not fully applicable in software and agile projects.

Project Management Body of Knowledge (PMBOK) is Project Management Institute's (PMI) Guide for project management and an ANSI standard. It identifies good practices and creates a common terminology for project management and is therefore widely used for project management trainings. PMBOK guide defines a project as follows: "A project is a temporary endeavour undertaken to create a unique product, service, or result" (2004, 5).

Project management refers to the application of skills, tools, techniques, and knowledge to meet the stakeholder expectations (PMBOK 2004, 37). Typically a project manager is the person responsible accomplishing project objectives (PMBOK 2004, 8). That is to plan, monitor, and report the different aspects of a project such as project scope, schedule, cost, status, quality, communication, risks, and organization (W2E).

Unfortunately, despite of well known project management practises only 35% of software projects were acknowledged successful according to Standish Group CHAOS report (cited in Hass 2007, 2), 19% of IT projects failed and 46% were over time and/or budget in the United States in the year 2006. Fortunately, the success rate has increased from earlier Standish Group research which states that only 28% of IT projects succeed fully, 49% were "challenged", and 23% failed (Fowler 2002). Thus, the situation with projects does not look good since in the Standish Chaos report, the success of the project is defined and measured according to success in on-time, on-budget, and with most of the expected features.

In a traditional project tasks are planned up front and they are completed one after another in an orderly sequence. The project usually has phases such as initiating, planning, executing, monitoring, and closing (PMBOK 2004, 8). Once a phase is completed, it is assumed not to be revisited (Lehtimäki 2006, 150). This type of project management is suitable for various and diverse projects, for example construction projects where all requirements and a

thorough plan for the whole building should be present before the basement is moulded.

However, in software development projects, this approach is not fully applicable; the requirements are not clear in the early phase of the project and they tend to change over time while the customer's comprehension increases (Hass 2007, 1). The ability to revisit the previous phases is called incremental and iterative development (Szalvay 2004, 4). Stepanek also (2005, 22-24) claims that software development projects are in many ways different from any other projects and even a capable project manager can therefore not guarantee a successful result with traditional project management. Software resembles more new product research and development than manufacturing (Szalvay 2004, 3).

Traditional project management practices lean on the assumptions that the project scope, cost, and time can be quite accurately defined and estimated in the planning phase (PMBOK 2004, 23). Stepanek (2005, 25) claims however, that these cornerstones of the project tend to change in software projects since software is fairly abstract, requirements do change often, and even software technologies can change rapidly during the project.

Project success can be measured according to the value it creates to customers. Another Standish Group study states that 45% of developed features were never used and only 20% of features were used often or always (cited in Ambler 2006). Agile methods target to **deliver value** to stakeholders **early** in the project lifecycle (Loeser 2006, 1). Agile processes allow changing requirements based on constant feedback from customer during the entire duration of the development project (Cohn & Ford 2003, 74). These abilities to deliver value early on and adapt to change are considered to be the major advantages of agile methods and the reason to their success (Loeser 2006, 2).

Traditional projects' stakeholders influence often decreases along the project duration and the cost of changes increases as shown in Figure 3 (PMBOK 2004, 21). The agile approach emphasizes the customer or user involvement

throughout the project (Schuh 2005, 11) to ensure feedback as early as possible and to avoid developing unnecessary features. Thus the cost of changes will decrease since need for them is detected earlier.

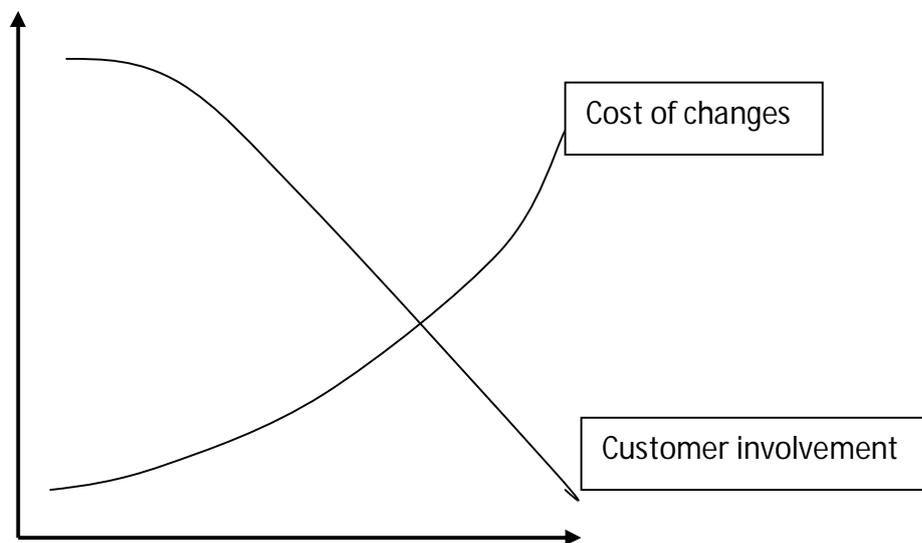


FIGURE 3. The influence of stakeholders and cost of changes.

Traditional project management involves disciplined and deliberate planning and control methods and has distinctive project phases (Hass 2007, 1) and aims to quality through application of a series of prescribed processes, documentation, and monitoring managed by the project manager (Loeser 2006, 3). Quite the contrary, agile project management aims to create value to stakeholders with iterative and incremental process where developers and project stakeholders actively work together to identify and prioritise needed functionality to satisfy the customer (Hass 2007, 3).

Augustine and Woodcock (2003, 7-14) have developed an agile project management framework which consist of following six practices:

- establishing and continuously reinforcing a guiding vision which supports customer's business goals (2003, 9)
- facilitating collaboration and teamwork with respectful atmosphere (2003, 10)
- establish and support team's guiding practices (2003, 11)
- open information (2003, 12)
- light touch in management (2003, 12)
- agile vigilance (2003, 14).

Currently, a number of different agile methods are used but according to recent Forrester research, Scrum is the most popular (West 2009a, 2); 84% of the studied organizations use it wholly or partially. Scrum is an agile project management method, which is argued to be a good starting point for changing the management practices to agile (Szalvay, 2004, 10). It does not define specific software development techniques instead it concentrates on how team members should function in order to produce software in constantly changing environment (Abrahamsson et al 2002, 27). It can, however encapsulate existing engineering practices (Schwaber & Beetle 2002, 57), or different new techniques, such as Extreme Programming (XP) and Test-Driven-Development (TDD), can be exploited and used with Scrum when necessary (Jacobson 2008).

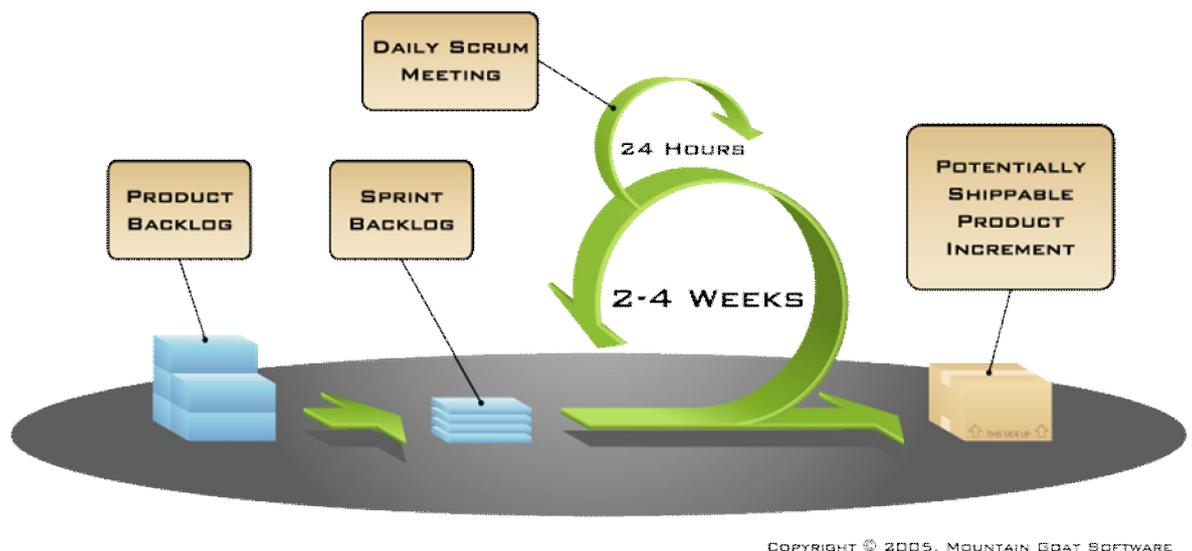


FIGURE 4. Overview of Scrum agile software development

Figure 4 visualises an overview of Scrum. On the left, the product backlog, which has been prioritized by the product owner, contains all requirements and change requests for the product that are known at the time. Next a selected sprint backlog for the 2-4 week sprint (iteration) which is illustrated by the larger green circle. On the right, the outcome of a sprint: a product increment that is a working piece of software. (Cohn 2005.)

At the start of each sprint, the team selects some amount of work from the product backlog and commits to completing that work during the sprint. At the end of each sprint, the team gives a demo of a potentially shippable product increment that is working, high-quality software, and evaluates the sprint to make improvements to practices. Each day during the sprint, team members meet to discuss their progress and any impediments to completing the work for that sprint. This is known as the daily scrum. (Cohn 2005.)

Although agile project management does not support traditional project measures such as estimated work load, workload left, estimated cost, etc. there are practices and measures how the workload and the productivity can be measured. One of them is velocity: how much the team can produce within an iteration. Traditionally productivity has been defined as output per worker-hour (Meredith & Shafer 2002, 47) but in agile the value is measured by summing up the “story points” estimations of the backlog items of the delivered working functionality in one sprint (Ambler & Kroll 2007b).

In addition to agile project management practices, there are significant amount of agile software engineering practices and methods that can be exploited to product development to improve the quality of code and to embrace collaboration within the development team. Agile development is an umbrella term used to describe a specific group of methodologies that arose out of a growing discontent with the way software development has been approached for the past 30 years (Schuh 2005, 2).

2.4 Software development

Software development projects have typically followed so called “waterfall” method and focused on software development life cycles over the last 30 years (West 2009a, 2). In this method, the project runs from business and system requirements through design to construction and test ending in delivery and maintenance. Each of these phases has defined input and output and several tools and techniques are applied to complete the processes (W2E). The emergence of agile development has brought several new

techniques to software development and the agile product management changes the previous “phases” mindset.

The waterfall model has been the most common way for large organizations to write software (Heusser 2006, 1). Companies try to turn the waterfall into an assembly line with requirements analysts, architects, coders, testers, and project managers who oversee the assembly (Szalvay 2004, 4). The projects have several other stakeholders, as well. Maintaining the appropriate level of communication and transferring information between these people is difficult, so projects tend to rely on detailed documentation (Heusser 2006, 1). Unfortunately, communication is never perfect and therefore vague or inadequate documentation can lead to misunderstandings, errors, or omissions. On the other hand too much of it takes valuable time and energy (Kusssmaul 2004, 126). So, managing imperfect communication creates therefore stress to projects (Cockburn 2007, 8-13).

Focusing on software development process has initiated process improvement and development of process models such as CMMI (West 2009a, 2). It is assumed that a well-managed organization with a defined engineering process is more likely to produce software that consistently meets the users' requirements within schedule and budget than a poorly managed organization with no such engineering process. CMMI process models do not contain process descriptions. Instead, CMMI provides a way to assess the state of an organization's ability to build software in a repeatable, predictable way. (Fritzsche & Keil 2007, 11-12).

Many agile methodologies were developed in response to ISO or CMMI qualified methodologies that emphasised detailed documentation and formal processes (Kusssmaul 2004, 126). Recent white paper from Software Engineering Institute however emphasises that CMMI and Agile methods can co-exist and be integrated successfully (Glazer, Dalton, Anderson, Konrad & Shrum 2008, 31). A lightweight CMMI based assessment is suitable for an agile context because it does not require documented evidence during the assessment and it seems to provide a mechanism to identify practical agile based improvement suggestions for the software development teams and

organizations (Pikkarainen 2008, 102). According to the study of Fritzsche & Keil (2007), most of the CMMI process areas can be fulfilled using agile methods. However, since some process areas, mainly those of the maturity levels 4 and 5 are in conflict with agile principles, agile methods can be applied without any major adaptations up to level 2 and up to 3 with some minor changes. (Fritzsche & Keil 2007, 25).

As already mentioned, software development projects are in many ways different from any other projects (Stepanek 2005, 22-24) and they resemble more new product research and development than manufacturing (Szalvay 2004, 3). Unique characteristics of software development vary from complexity and abstractness of software to immature technologies. Great deal of this uniqueness is also due to rapid changes in requirements and used technologies as well as the vagueness of the final result. Software product is rather designed than made and the outcome is abstract. (Stepanek, 2005, 8-22)

Requirements emergence during the development project is incompatible with past process practices (such as requirements-driven sequential waterfall process model) and with process maturity models emphasizing repeatability and optimization (Boehm 2007, 712). In their place, more adaptive and risk-driven models are needed. An adaptive development plan is needed because of the uncertainty and significant amount of unknown variables (Szalvay 2004, 4) and development teams must concurrently develop flexible processes (Anderson 1997).

Furthermore, the criticism of waterfall method also includes the claim of diminishing customer involvement along with project progression, superfluous documentation, and using time for developing unnecessary features (Lehtimäki 2006, 151-152).

Agile development itself is not a methodology. Instead, it is a set of fundamental principles how software should be developed (Schuh 2005, 2). It is said to create a framework of methods, practices, and processes of software development (Cohn & Ford 2003, 74). "Agile practices are the things

that agile teams do every day to write quality code, deliver useful features, plan and track progress, and react to change” (Schuh 2007, 61). The traditional project management methodologies were designed by managers to control projects whereas agile methodologies were designed by technical community. Hence agile methods focus mainly on the development process. (Augustine & Woodcock 2003, 6.)

What different software development methods have in common are identified as agile characteristics: software development is incremental (small releases, with rapid cycles), cooperative (customer and developers working constantly together with close communication), straightforward (the method is easy to learn and to modify), and adaptive (able to make last moment changes) (Abrahamsson et al. 2002, 17). Key agile principles according to Augustine (2006) are:

- focus on customer value: the features are prioritised according to business value they will bring
- iterative and Incremental Delivery: customer receives a “flow of value” by incremental deliveries
- intense collaboration: co-located teams which are encouraged to constant knowledge sharing and collaboration having wide range of competence in the team
- self organization: sharing a guiding vision the team spontaneously organises and directs towards it in best possible way and
- small, continuous improvements: teams reflect, learn and adapt to change.

Currently, the recent Forrester research states that Extreme Programming is used in 38%, iterative development in 47% and waterfall is yet used in 33% of the studied organizations (West 2009a, 2). Being relatively popular there already is some experience of how agile product management and development techniques change the management style and practices.

2.5 Agile management

While agile development teams drive to be self-managed, pull work items for themselves from product backlog and strive to improve their processes and methods frequently in the end of each sprint, the role of management becomes quite different from before. Since software engineers or “coders” are well paid employees they create significant cost to business and thus their contribution should be maximally exploited (Anderson 2004, xxvi). To accomplish this, skilled management and reliable metrics are needed. Even human resource practices may need to be changed. Our sense of the world dictates the management style (Highsmith 2004, 19); if the world is perceived static – the management style follows the early planning and minimal change pattern. On the other hand, if the world is perceived dynamic – nominal early planning is followed by ongoing learning and change.

Stepanek (2005, 131) argues that doing iterative development will change the management practices dramatically in order to get successful results and the earlier traditionally well regarded as good and best practices do not therefore apply any more. Agile project management requires more and better leadership skills; keeping the spotlight in the vision, inspiring the team, promoting teamwork and collaboration, etc. than just creating a plan and monitoring and controlling the project (Augustine & Woodcock 2003, 6).

Ambler and Kroll (2007a, b, and c) introduce lean software development governance practices which mainly focus on encouraging the use of collaborative strategies in order to enable and motivate development team members implicitly. Instead of loading the development team with heavy rules, guidelines, processes and accurate development plans there is minimum set of external control present. Creating a guiding vision, encouraging teamwork, collaboration, and open information, setting only simple rules, and having a light touch and agile attentiveness in monitoring and adapting are enough (Augustine & Woodcock 2003, 8). The agile approach in projects spread responsibility to team members. Everyone has more planning orientation than plain executing the tasks handed out by project manager (Palomäki 2008, 19).

Hass (2007, 4) states that typical key elements used in agile project management are the following:

- visual control in means of a burn-down chart or other cards-on-the-wall planning method for organising the work
- co-located high-performance team, preferably in one team room to increase coordination and communication
- test-driven development, which may realise in developing the test cases before finalising requirements
- adaptive control is needed to lead constant adaptation to improve working methods
- collaboration with customer in defining the requirements and getting feedback as well as in design, development, testing and reworking within the development team
- feature-driven development which allows the team to focus on one feature at a time
- leadership and collaboration rather than command and control
- move from cost to revenue focus when prioritising features by value
- lessons learned by exploiting frequent retrospective after each iteration.

Leadership in agile projects and development teams value the individuals and interactions over process and tools as stated in The Agile Manifesto (2001). Key characteristics of a successful leader include ability to know when to take the lead and when to let the team to decide how to get the job done. This will require courage and self-awareness since the team will be eager to experience propositions that ignite their passion if allowed. If the team is inspired to be innovative and creative, thus focusing on delivering value, they will continuously reflect upon their performance and change their practices for better. (Pollice 2009.)

Usually agile development is scaled to manageable level using small teams (Augustine & Woodcock 2003, 5) which are located in the same premises and preferably they could have a shared project room. (Hass 2007, 4) Since agile teams value face-to-face communication, the leader must want to listen and communicate with the team members spontaneously more than require formal meetings and status reports (Pollice 2009).

Hence the change is significant problems may arise in agile implementation. Apart from the “normal” resistance for change, problems can include and show up in various ways. For example, difficulties in planning and communicating road maps and portfolios, not enough time for long term planning, good documentation practices disappearing for a moment, weekends are used to finalise sprints, only Research & Development works in agile incremental way, other stakeholders use old linear model (Palomäki 2008, 20).

Scrum and XP may raise resistance from management since document based evidence of progress of the projects may not be available (Gustafsson 2008). However, while iterative development creates frequent demonstrative outcome and incremental value to business (Ambler & Kroll 2007a) it also creates possibility to early risk reduction and control points (Ambler & Kroll 2007b). Thus short iterations with demonstrations of deliverables in the end provide concrete evidence of the project progress they more reliably show the status than document based workload calculations (Ambler 2008).

New metrics for measuring the economical and business consequences of agile product development are needed. Anderson (2004, 49) suggests that calculating the quantity of production output against the input that is the ideas against delivered features, and throughput time will produce more reliable measurements than traditional ones, such as man-hour estimations per new feature or calculating lines of code, did. The production throughput creates a flow of value which has to be managed – not the single activities, as before (Augustine 2007). Moreover, aligning payment, incentives, job titles, promotions, and other recognition with agile by creating incentives promoting teamwork and shared responsibility (Keith & Cohn 2008, 28) has significant effect on HR planning.

In today’s changing world, business runs short-term cycles, requiring frequent change, innovation and improvement (West, 2009a, 2). A company could benefit of agility instead of doing everything in the traditional way. The

business model and internal processes should therefore reflect on the agility used in product development.

The operational environment of the company become increasingly complex and the velocity of the transformation increases. The more complex, interactive, and transforming the environment becomes the more agile the company has to become in its strategy. Leadership that is based on knowledge and thinking in the organisation will create sensitivity in strategy and thereby ability to manage the complexity and changes. (Doz & Kosonen 2008, 261.)

Traditional business management usually consists of three year strategy plan which is annually revised and an annually produced action plan which is followed-up regularly and sometimes changed during the year. Targets based on action plan are assigned to teams and individual workers mainly in a development discussion which is held once or twice a year. Financial forecasting is usually today done annually but it is a rolling model so it is revised monthly. Product portfolio road mapping and technology road mapping are done for three next years and they are revised annually.

Strategic agility is opposite to periodically repeated procedures which lead to rigidity and eventually to crisis. It needs sensitivity in strategy, flexibility of resources, and collective commitment since it means the company's constant ability to make accurate real time perceptions, commit collectively to decisions and to allocate the resources rapidly and in sufficient scale. (Doz & Kosonen 2008, 33.)

Changing the working culture and learning new ways of working takes more time and is more difficult than introducing and teaching agile methods (Palomäki 2008, 22). Strong management is critical to the success of adoption and application of agile methods (Augustine & Woodcock, 2003, 3). Thus selling the changes especially to project managers and middle management is crucial for a successful rollout (Ambler & Kroll 2007b). Everyone has to commit to the new methods and top management support is also essential (Palomäki 2008, 22).

2.6 Organizational changes

Introducing change into an organization is frightening and difficult. Agile development is even more difficult since it says: first, change the way you do things, next, keep changing, forever. Knowledge and experience is needed, change has to be planned and the learning process and different learning styles have to be considered in order to get the commitment to the new and to embed it deeply to mundane life – not just to the speeches and to the process intranet.

Expectations of rapid change are not always realistic. Satir studied people's reactions to change. Her key observation was: "if there is ever a question between comfort and familiarity, familiarity will almost always win out". Even if the familiar mode is inefficient and uncomfortable, people stay with it (cited in Cockburn 2007, 319). Unfortunately, besides the force of habit personal goals may also clash with agile principles and practices: current status quo or career ambitions as well as work-shyness may decrease the enthusiasm to adopt agile roles and responsibilities (Keith & Cohn 2008, 25).

Since there is a promise that a company will achieve breakthrough results by making many small changes and reflecting on those changes (Cockburn 2007, 323), many organisations take the risk and learn agile methods. New competence is needed, when a company wishes to acquire new processes and methods. This competence can either be bought or "stolen" that is, copied from others (Kettunen 2008, 62, 1).

Information/knowledge can be fully exploited or transferred to production after four phases:

- deep concentration on the information
- implementation of the information/knowledge
- assurance of the implementation and
- widening the application of the knowledge (Kettunen 2008, 77).

Top management's command or forcing may not be the best approach for implementing new methods although external threat helps in instigating

change (Cockburn 2007, 326). This however may turn out to have undesired consequences. When new ideas tend to come from within the organization rather than from an outsider creates the “we are tuning it for us” principle – a critical success factor for the adoption of new processes, says Alistair Cockburn. (Cockburn 2007, 323.)

Migrating iterative development needs resources and creates costs to the organization in many ways: training and mentoring of new way of working are needed, demand of broader skill set for individuals emerges, testers are needed earlier and architects longer, project management requires higher degree of involvement etc. (Ambler & Kroll 2007 b). To perceive and assure everyone of the benefits of endeavours and the new way of working measures should be taken. Identifying common metrics takes time but once established these metrics are used to increase collaboration, the “we are all us” mind set (Cockburn 2007, 326). However, trying to reach too many objectives all at once disturbs the exploitation of new knowledge (Kettunen 2008, 77).

According to Cockburn (2007, 14-15) people who are learning and mastering new skills pass through three quite different levels or stages of behaviour: following, detaching, and fluent. People in the following stage look for one procedure that works. Even if 10 procedures could work, they cannot learn 10 at once. They need to learn one first. And they learn it by copying it.

In the detaching, or level 2, stage people realise the limitations of a single procedure and look for rules when the procedure breaks down. The person in the detaching stage learns to adapt the procedure to varying circumstances; he is now more interested in learning the 10 alternative procedures, in learning when each is most applicable and when each breaks down.

In the third, fluent stage, it becomes irrelevant to the practitioner whether he is following any particular technique or not. His knowledge has become integrated throughout a thousand thoughts and actions. It does not matter to him whether he is following a procedure, improvising around one, or making up a new one. He understands the desired end effect and simply makes his way to that end. (Cockburn 2007, 14-15.)

Forrester Research has found in their research that successful adoption of agile methods in software development needs the application of lean principles across the company to overcome many cultural and organizational problems. The lean approach encourages replacing traditional techniques with ones that reduce waste and increase value for the customer. Lean principles change how business is planned and projects are measured. It ensures that only the right amount of planning is done at the right time. The lean principles can be seen in reality in the following organisational characteristics:

- processes are simpler
- customer involvement is more natural and
- organizations are flatter. (West 2009b.)

“Organizations that have taken advantage of both agile and lean approaches together have seen benefits such as reduced costs, improved time to market, and higher quality. But perhaps the most surprising result is in the areas of innovation and staff motivation. These organizations are highly motivating places to work, with team members feeling that they're contributing to the company's bottom line.” (West 2009b.)

Sutherland (2008) gives guidelines how to change software development into agile and take the best out of it:

- Firstly, Scrum practices have to be implemented and “the Nokia test” passed. The Nokia test is a simple questionnaire asking whether the team has time-boxed sprints lasting less than four weeks, delivers fully tested working software after each sprint, starts sprints with agile specification, has a nominated Product owner, and has Product backlog etc.
- Secondly, management has to be totally involved and understand velocity of the team and remove obstacles.
- Thirdly, test-driven-development and continuous integration are XP practices that have to be implemented.
- Finally, the development team may reach a state of hyperproductivity that is five to ten times increased productivity than before (and more).

2.7 Agile tools

Agile practices are meant to be adopted and adapted by teams in order to help them produce better, more valuable functionality faster. Tools also can help do things better and faster, but tools are not as malleable as practices. It is therefore recommendable to adopt practices before tools to get really agile (Schuh 2007, 65-66).

Distributed agile software development contradicts with the original agile methodologies which have their roots in small, co-located teams (Frye 2007). However, distributed teams are reality in today's global markets and global workforce and organisations adapt agile methodologies to meet the quality, productivity and efficiency expectations. It is therefore necessary to exploit specific tools to fully utilise all distributed team members. In distributed teams' environment, face-to-face communication and collaboration is not always possible in real time due to time zones and diverging holidays (Kussmaul 2004, 125). The teams can benefit of advanced tools for planning, communicating, and collaboration in general.

Visual Studio Team System (VSTS) is a tool for all the people working as developers, testers, architects, project managers, and product owners to provide them with a software project environment for the product development (MSDN). In VSTS, the product backlog is maintained, software code is stored and the configuration managed. Work items can be tracked from initial requirement all the way through sprint planning, execution, testing to delivery.

Modern technology provides tools for both planned meetings and instant communication needs with VOIP phones and Microsoft Office Communicator chats and live meetings with application sharing. Even web cameras can be used to get a feeling of being in the same meeting.

2.8 Remarks of the literature review

Although most of the literature emphasizes the positive outcome of implementing agile methodology or even some agile practices to software development and product management there are opposite opinions, too.

Abrahamsson (2008) gives ringing claim that agile suffers severely from three perspectives: conceptual confusion since the terminology around agile is incoherent, empirical implementation has no reference framework to support tailoring, and philosophical interpretation due to the lack of support for composing "an agile process", "anything goes" as long as it is "agile". However, such attempts to create agile adoption frameworks do exist, for example Pikkarainen, Salo and Still (2005) have done one in a case study on F-Secure Corporation deploying a new agile software development process.

Not every project benefit from adopting agile practices (Schuh 2007, 43) although there is impressive evidence of success: for example in 73% of embedded software projects using agile methods were considered successful and the outcome was either positive or very positive in a ITEA-study in 2006 (Abrahamsson 2007).

Jacobson (2008) claims that different processes, technologies, and methodologies become fashionable since organisations are searching a silver bullet to solve all problems in software development. "Agility is not the cure-all for our software process pains" (Gustafsson 2008, 1). Therefore, implementing agile will fail unless the value chains of the business are identified and it is ensured that with agile the company can deliver value to market and respond to market changes faster (Cottmeyer 2009).

The value-based approach provides a framework for determining which low-risk, dynamic parts of a project are better addressed by more lightweight agile methods and which high-risk, more stabilized parts are better addressed by plan-driven methods. Such synthesis is becoming more important as software becomes more product-critical or mission-critical while software organizations continue to optimize on time to market. (Boehm, 2007, 712-713.)

Burke (2008) states that currently there is a second wave of implementing agile. Earlier small software companies adopted it but now "the chasm" (Moore 2005) has been crossed and larger and different types of industries and big enterprises are also adopting it. Burke also argues that agile can be

used in many different projects and organisations: even sales or marketing can work in two week sprints. (Burke 2008.)

3 AGILE METHODS AS A TOOL FOR EFFICIENCY

3.1 Expectations

The basic goal of agile methodologies is to help developers to create better software more easily (Stepanek, 2005, 66). Following the basic practices of agile management framework will add value to projects and also make the achievements and making of them enjoyable (Augustine & Woodcock 2003, 15).

The corporation has a well defined project management process to create systematic, consistent, and successful projects in all areas. However, for product development the templates, the decision making process and the project organisation seemed to be too exhausting: for example, the empty project plan template has 11 pages, the requirements specifications template 12 pages, and normal status report template four pages. The mandatory metrics for measuring project success also raised questions – they were not always considered applicable.

The decision making process was too distant from the development project perspective: decision making should be a regular team activity rather than single decisions in some distant steering committee. However, the project management process does have many advantages, too. There are very good templates, checklists, process activity descriptions, and examples available in the process intranet.

In Business Unit xxxxx there is reliable evidence of advantages of adopting agile project management and development methods in one country. In Norway, they have created a fully functional agile development team from a chaotic team which followed no definite production model or development process. The team was unfocused, inefficient, and unproductive with unbalanced responsibilities. Currently, the quality of the products has

increased to respond to customer expectations measured by both functional correctness and reasonable defect rate. The net sales and the growth of the business show good performance and the situation in overall is stable, there is control and responsibilities in right combination.

In other countries, the problems with time-to-market, time-to-react, efficiency, productivity, and quality urged to change the product development process. These were the compelling problems identified in most teams and functions throughout the organisation and improving them was set as objectives. The experience in Norway and the overall popularity of agile methodologies had convinced top management and agile methods were therefore considered an option worth trying. According to the quality and process manager, saving money in product development was not included in the objectives, although cost-effectiveness is not regarded as an unfavourable consequence, either. Finding errors as early in the development process as possible, would decrease their cost effect hence testing was raised as one issue for improvement.

Better communication, collaboration, new processes, and using resources more flexibly were considered desired benefits, too. Some were expecting agile methods to be more fun to work with; some plainly wanted more modern processes. There were some, who had earlier knowledge or experience of agile and they expected Scrum-methods, daily stand up meetings, sprint planning, and demos to increase communication and collaboration in team.

Many people in different teams, functions and management levels shared most of the described expectations. However, a realistic view of changing the processes slowly, gradual step by step improvements for the better, were seen as the means to achieve these expectations.

Ability to react quickly and rapid development cycle in responding to customer needs and law demands would give the customers an impression of a competitive, capable and flexible company. When a market window opens the implementation of new features has to happen in three months in order to be

profitable, otherwise it is too late. Earlier the development cycle was one year and therefore the product development had to be reorganized.

3.2 Organising product development

Agile principles are used in the development and maintenance of our product portfolio. Exceptions are only allowed by development manager depending on local circumstances. The portfolio consists of several product lines which are owned, managed and maintained by Product Line Teams (PLT:s). The PLT includes everyone involved in the work on a product line. The work is organised to boards, councils and teams which have stated roles and responsibilities.

There is one Business Board per product line such as, Elderly care, Family care, or Education. The members in this board are country representatives and development representative. The responsibility of Business board is to act as an owner of the assigned portfolio. That includes several activities such as, creating a product line roadmap annually and revising it every quarter, creating an overall release plan (schedule and desired content) annually and revising it in every quarter, prioritising of resources quarterly.

Each product, such as children day care classic, children day care web, and learning platform, has a Product Council. The members in this council are Business Board representative, Scrum master, and specialists (architects, domain experts, sales and support representatives).

The responsibility of Product council is:

- to make detailed release planning that is, all deliveries are scheduled in frequent releases and service packs
- to manage the Product backlog that is, requirements stabilization, prioritization, and ranking and
- coordination with other product councils within the same product line.

Product council also acts as a steering group for product development and release projects.

There are one or several agile teams in which the development of a product is done. In the team, there are all resources and roles involved in the development work (all phases) for a suitable part of the product. The fundamental expectation for a development team is to deliver piece of working software after each iteration. The responsibilities of the agile team include: frequent sprint planning, designing, coding, testing, error fixing, and inter-project communication. The Scrum master is acting as project manager.

Figure 5 visualises the annual, quarterly and sprint cycles of product line work in each PLT level. Once in a year, a major release is published according to the road map done by Business Board. Within a year, several planned minor releases and unplanned hot fixes will be released according to Product council's plans. The actual work of these releases is carried out in Agile team sprints.

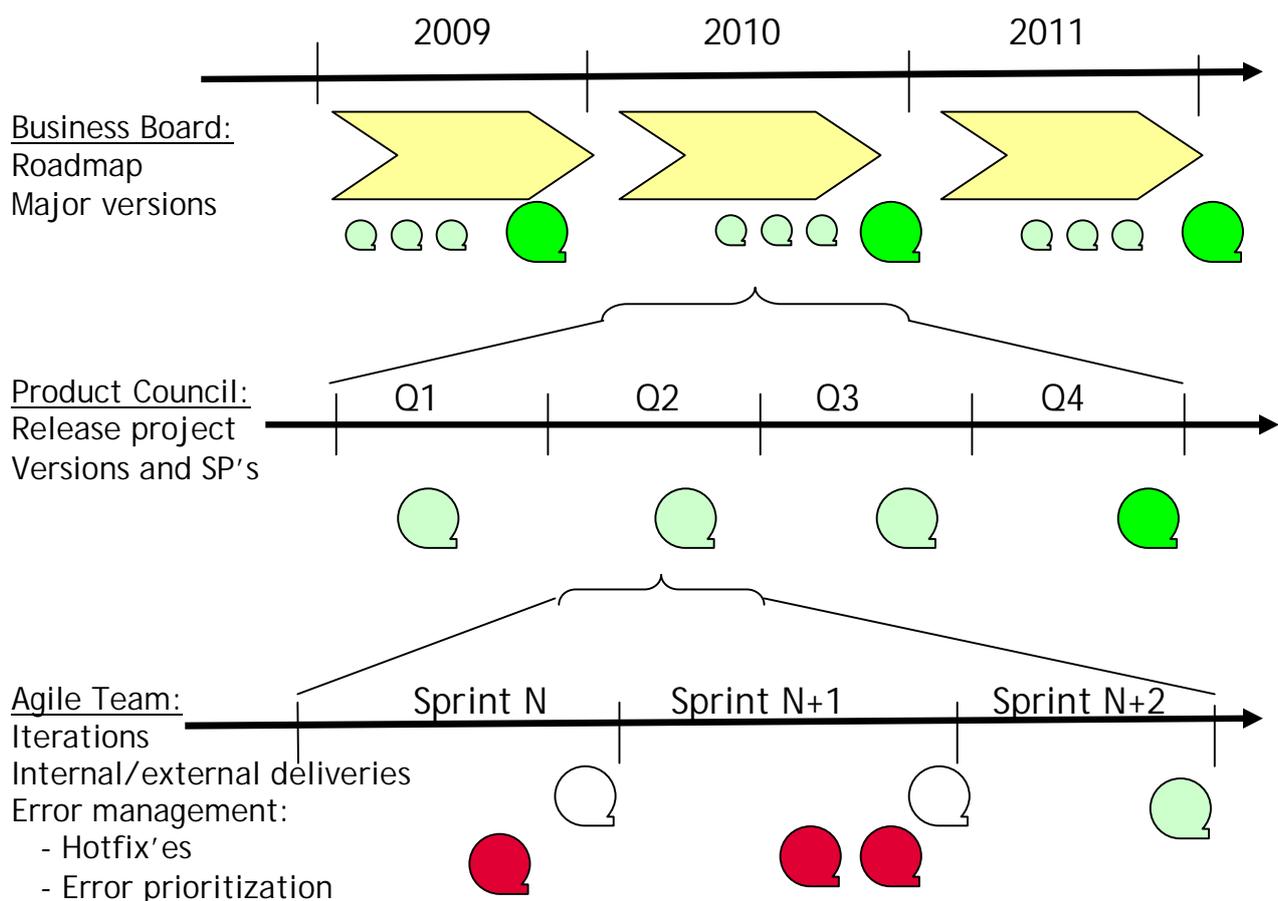


FIGURE 5. Product Line Teams: Release pattern

In addition to earlier mentioned teams, there are Specialist teams which have specified roles and responsibilities such as, configuration management, building, test environment deployment, validation testing (=acceptance testing), packaging and deployment (delivery to the customers).

3.3 Product lifecycle

The product lifecycle describes the life of the product from an idea through the mature product to decline. The product lifecycle in Business Unit xxxxx has seven main phases, which all products and enhancement features pass at different time and different speed:

1. Business case: Clarify customer need, customer value and revenue expectation
2. Solution concept: Describe offering in customer domain terms
 - High-level use-cases / stories
 - Input – Output – Other results – Constraints – First cost estimate
3. Architecture: Describe how the solution fits within the offering portfolio
 - Main components, functionality and interfaces, revised cost estimate
4. Development: Produce a working product (possibly in stages)
5. Delivery: Assemble a complete deployment package (including sales material and other documentation)
6. Maintenance: Collect, prioritize and implement changes and improvements
 - Miniature version of main flow
7. Retirement: Remove from deployment packages, clean-up data, update documentation and collateral.

Most of the products are in the maintenance phase but new additional features and customer specific adaptations are frequently released. All of these enhancements pass through the lifecycle in different time and at different speed. Significant effort is spent on categorising, evaluating, and prioritising the product change requests and new feature request for selecting candidates for development in different levels of Product Line team

organisation. Selection and ranking are mostly done in the grounds of return on investment (ROI) and customer satisfaction.

3.4 Roll-out project

Schwaber (2007, 9-11) gives some recommendations how to start with Scrum. In his opinion, it is important to start immediately by establishing an Enterprise Transition team, holding a kick-off meeting, and starting some pilot projects and not to plan too extensively. Additionally he emphasises that Scrum should not be modified. Instead it should be implemented as it is and if changes are needed they should be directed at the company processes to fit them with Scrum. Schuh (2005, 46-48), quite the opposite, encourages to start adopting just one or few agile practice into a project and consider whether the best way to start is doing one agile methodology by the book or modify or partially implement one or more methodologies.

In any case, Cohn (2008) emphasises that the organisation has to make deliberate decisions whether to:

- start with small changes or all-at-once attitude
- start with engineering practices or just iterative development cycle or
- publish the roll-out immediately in the beginning or only after some successful pilots.

Conscious consideration and choices will improve the prospects for successful transition (Cohn 2008).

Since our vice president was fascinated of the remarkable advances accomplished due to implementation of agile methods in Norway, he examined the case and acquired more knowledge of agile methods from literature. Then he together with the quality and process manager and the development manager decided to inject some agile practices into few pilot projects. As Scrum was the methodology used in Norway that was chosen to be the starting point in other units as well.

The pilot projects had a positive attitude towards experiments on agile. In one of the development teams, an experienced scrum master was employed and

he acted as mentor to the other teams. An external agile coach was invited to share knowledge and experience with the development unit's employees. His practical stories convinced many from the development unit of the superiority of agile compared to traditional methods.

Agile practices are learned through example, application, and training (Heusser 2006). Recruiting new people is seen as a possibility to influence in the ability to change and develop because new people bring new information (Kettunen 2008, 101).

In the roll-out, Scrum roles and practices were adopted according to Scrum checklist which the external coach had provided (see appendix 2). The Scrum team role was occupied by development teams, coders, designers and architects. Some of the old project managers were turned to Scrum masters. In development unit, Product owners were appointed to each product but basically the ownership of the product and product backlog is in PLT.

Each development team was responsible of the adoption of the new methods in learning by doing and they started with different artefacts and pace. Management was supporting but did not interfere to the implementation. Different stakeholders of the teams, such as product owners, product managers, PLT governance bodies, line managers, testers from customer support, sales, and consultants, learnt the new way of working through experience. Some general trainings were held, but not to all organisation.

Learning is a dynamic concept that refers to continually changing organisations and can in particular be seen as a change in behaviour (Maula 2006, 13). Experimental learning has cycle of four stages and people learn in variety of ways. These two assumptions materialised in the different roles of learners and stages of experimental learning: activist (having an experience), reflector (reviewing the experience), theorist (concluding from the experience), and pragmatist (planning the next step). Knowledge of learning helps to become more efficient in it. (Foot & Hook 1999, 174.)

The stakeholders outside development team especially felt that the implementation was done by trial and error, the roles and responsibilities are still unclear and the roll-out of VSTS tool at the same time was confusing since no training was provided. They were not quite sure how and when they should participate in different Scrum activities. Despite lack of involvement or training, innovative applications of agile methods have been done outside development unit; first line support has started daily stand-up meetings with second line support and managers have frequently short status meetings with “actions backlog”.

Learning requires conflicts between old and new knowledge; the new knowledge clashing to the existing knowledge challenges productivity and facilitates learning (Maula 2006, 17) so when learning new the memory is disrupted by old and new issues (Kettunen 2008, 65). Unfortunately there are also individual barriers to learning such as, people do not see the importance of learning or change, they do not have time for reflection, they doubt whether they are able to learn, or they are not interested in putting the theory in practice (Foot & Hook 1999, 177).

In Scrum, there are only three roles: Product Owner, Scrum team, and Scrum Master (Schwaber 2007, 106). In the development unit, there were several people who did not fit to any of these Scrum roles; product managers, line managers, project owners and managers, test coordinators, second line support people, etc. Some of them felt confused of their role in the new agile world. Others questioned their roles as well.

Reinforcement (praising and punishing as behavioural response) and feedback are particularly important to learn effectively and to develop, for example trainer’s comments on progress, manager’s appraising the work of their staff etc. (Foot & Hook 1999, 173).

In conclusion, the transition from traditional project management and software development to agile will have an impact on substantial amount of current organisation, processes, roles, methods, practices, techniques, and tools. At

least, the following has to be considered whether any changes are needed or not:

- decision processes
- reporting procedures
- product management process, organisation, roles, and methods
 - o product data management tools: VSTS, OMT
 - o requirements development and management
- project management process, organisation, roles, and methods
 - o scope and results
 - o schedule
 - o cost of project
 - o work load estimations
 - o monitoring the work
 - o work load and concentration on project
- development working process, roles, and methods
 - o quality assurance
 - o configuration management
 - o testing
- communication and information sharing
 - o communication tools: office communicator, Teamer, Project office

Utilising agile methods may not have an effect on everything, but gradually these issues have to be considered.

In some areas, improvement is however needed

- management should get more experience of agile, perhaps they could be involved in an agile project
- good practises should be shared between different teams, for example Scrum masters could have frequent meetings for sharing practices
- the roll-out concentrated mainly on development unit, the country units need more education
- the Scrum masters need more education and perhaps also the Product owners
- the roles and responsibilities should be clarified to everyone in development unit and to every stakeholder

- the structure of the organisation should be considered, most efficient would be that one team would have the “whole” responsibility of a product all the way through to customers
- common training for everyone, selling the benefits to the whole organization!

3.5 Means: practices and tools

Agile frameworks include numerous practices, tools, and ideas how to execute more agile software development that is, “all agile implementations are, by definition, adaptations of concepts, ideas, techniques and practices of agile family of ‘stuff’”(Abrahamsson 2008). Kniberg (2007, 7) emphasises that in every specific situation this framework has to be adapted that is, there are no clear guidelines to follow to guarantee an infallible success. Scrum has no process descriptions. Instead it employs the empirical process control model which adapts activities to produce desired and predictable outcomes (Schwaber & Beetle 2002, 100). However, trying to exploit too many objectives simultaneously may disturb the endeavour (Kettunen 2008, 77).

Product backlog is an important artefact in Scrum. It contains the user stories for required features. Visual Studio Team System is used as a tool to store and maintain the product backlog and coordinate the workflow of development work items from product owner to all stakeholders for example to developer and tester. Currently, in VSTS a CMMI development model based template is used; a Scrum template should be taken into use to be able to utilise agile features, for example burn-down chart.

All requirements, change requests, test items, and bugs are added into VSTS as work items. In a work item, there is a user story and requirements, coding and testing information all combined. The Product owner is responsible of updating the prioritisation and ranking of the work items according to PLT decisions. The Scrum team is responsible of estimating the workload for each work item. From product backlog the Scrum team takes items to their sprint backlog and commits to develop them - and only them – in the sprint.

Scrum disciplines: sprint planning, daily stand-up meeting, demo, and retrospective are used. In sprint planning, variety of tools are used for prioritisation, estimation, and ranking of user stories. These tools include for example MOSCOW – Must have, Should have, Could have, nice to have (Wishes); Scrum poker; pointing with fingers; T-shirt estimation.

Different communication tools such as office communicator, live meeting, and teleconferencing are used. In distributed teams, web-cameras have also been utilised. For information sharing an MS Share Point Server solution is used for storing project data and product documentation. Internal wikis and blogs are used for sharing topical information within and between the teams and also with other stakeholders. Synchronous communication methods, such as face-to-face meetings, online chats, and teleconferences, are ideal for quick status meetings, brainstorming sessions, and reviews (Kussmaul 2004, 127).

There are software engineering practices that could be utilised such as pair programming, automated testing, test driven development, and code reviews. For example automated testing combined with continuous integration makes it possible to release often. Without automated testing and continuous integration, frequent releases create a huge manual testing burden that is often unbearable (Heusser 2006, 1) therefore fully automated regression test is recommended to be able to ensure fully working product after each iteration (Cockburn 2002, xxii) which is a fundamental criteria for doing agile development (Sutherland 2008).

Typical agile project status report according to Cohn and Ford (2003, 77) includes: “a list of key dates, a two to five-paragraph commentary of the project’s state, a burn down chart comparing progress to planned work, key metrics (defect inflow, percentage of tests passed, and so on) appropriate to the project’s current state, and a list of key risks.”

Moreover, requirements development and management should be fine-tuned: the user stories must fulfil minimum requirements to be useful. The project management cannot continue using the old waterfall type of approach in producing specifications and requirements since the development team

cannot guess the priorities and rank the work items by themselves for two to four week iterations. However, this may also result a situation where the product management is not able to produce requirements at the pace the development team develops them (Cohn & Ford 2003, 77).

Teamwork has many challenges and some of them are emphasised in distributed and offshore teams. Sometimes projects are forced to be distributed in multiple locations: the business representatives or individual team members cannot be co-located, costs are lower in certain countries et cetera (Jens 2007). Strategic, communicational, cultural, and technical aspects concerning requirements and architecture, access to expertise, team dynamics and even security can each inflict surprising clashes (Tadipatri 2007). In addition, time zones and national and religious holidays challenge real time communication and collaboration (Kussmaul 2004, 125).

A word of warning from Heusser (2006): “It’s easy to think you’re doing agile right, and be wrong.” Throwing away old methodology binders and requirements and not replacing them with anything, for example with automated testing, communication, and feedback, does no improve the development process for better. “Without direction, a team told to throw away its waterfall method will simply devolve into ‘code and fix’. That isn’t Agile – it’s chaos.” (Heusser 2006, 1.)

4 AGILE METHODS IN USE

The literature presents different approaches to agile adoption in software development. On one hand, Schwaber (2007, 9) emphasises that the change begins and is managed by the top management who form the first agile team to lead the transition in the entire enterprise. In addition, two other types of teams are needed: roll-out teams which are responsible for the adoption work and actual development teams using agile principles. Schuh (2005, 13-14), on the other hand, encourages any team or project independently to try out agile practices and if they encounter some problems with the environment, that is

customer, stakeholder, and management, strive to alter the relationships for better if possible.

In Business Unit xxxxx, some pilot projects were run and based on the positive experiences; agile principles were implemented in all development teams during the autumn and winter 2008-2009. Trainings were held and teams improved their practices according to reflections and assessments done after each sprint. Some achievements can already be seen, however transition will take more time to earn out. The entire organization has identified or at least heard of some changes in the development process. Furthermore, some innovative application of agile practices have been exploited in other teams as well. Nonetheless, the change does not seem to have had major impact on management processes, yet.

4.1 Achievements

Following the basic practices of agile management framework will add value to projects and also make the achievements and making of them enjoyable (Augustine & Woodcock 2003, 15). “Software people do not like to see software engineering done unsuccessfully, and try to make things better” (Boehm, 2007, 698).

In Norway, where agile methods have been used for some years now, quality is not an issue any more; the situation is stable and controlled. Their achievements are tangible: they have the best quality and productivity, the net sales and growth are in good level. More abstract achievements include open atmosphere and commitment: “everyone is in the boat.” However, not all changes may have resulted in agile implementation, since they got a new manager at the same time, as well.

In other countries and product lines, the achievements vary from delicate hope and assumptions to concrete change of behaviour, still some are sceptical if any progress has yet happened. Adopting agile practices has been the most significant change for individual employees for years. Thus the most important achievement at current stage may be something many mentioned: people are

enthusiastic and committed to agile implementation, working in an agile way, and continuously reflecting to find areas for improvement.

Everyone knows that development teams are working with agile methods and they even identified Scrum artefacts such as daily stand-up meeting, sprint planning, demo, and product backlog. The organisation and roles are more structured and the collaboration and communication between country units and development teams has increased significantly. In addition, country unit feedback, from persons who are responsible of the requirements to development has increased. Work items are constantly prioritised. Flexibility and time-to-react have improved. For latest release even last minute requirements were met in one product! While the most important issues are dealt first, the delivery brings more value. The work can also be predicted more easily by viewing the product backlog and VSTS as a tool for product backlog works fine.

Currently, the testers participate in each sprint and it is considered to be an advantage. Even some improvement of quality of products was detected but it was claimed to be due to increased testing and getting testing resources more flexibly when needed not due to agile implementation. Regardless, since testing is more flexible than before during the sprints, the development cycle is faster, the verification test by product owner succeed more often, and there are less critical errors in customer environments. In addition, customer support or consultancy know better what is under production and what is coming to deliveries since they participate in testing more intensely.

The quality of code and products in general are assumed to improve gradually, latest release after agile roll-out seems to have good quality but there has not been any follow-up data available to show that. Anyway, improving quality makes customers, customer support, and even the developers happier. The developers are especially happy, since they feel that they are able to achieve more and keep the schedule better than before the agile sprints. "The people around me are growing, they take responsibility, and they flourish!"

Agile brings transparency which exposes problems. In addition, the work becomes visible and inefficiency cannot be hidden. Individual responsibility has increased both in development unit and in support unit. Teams discuss problems more openly; problems are more visible than before. People can deal with problems and not to be afraid of asking help from others. Problems are more often solved inside the team. "Daily meetings are great; it creates the team spirit and you get more help from others."

In development teams, the new process is evolving; people learn to work in agile way, which is continuously improved. In some teams, however, the improvements are left for individual responsibility, but in general teams are able to change their way of working if they want. They are freer and that results in commitment from team members and they feel the team spirit. "I feel that they have more fun at work and that will be of benefit very soon."

Supposedly, a great deal has already been achieved otherwise the roll-out would have been stopped. Today there are no achievements to show to customers, yet. Questions of analysing the benefits and achievements were raised though; measurements and concrete evidence are demanded to be evaluated. Cohn (2006) points out that reasoned measurements should be carried out: measuring only what can be measured, measuring in correct level, and measuring only something that will be acted upon.

However, we are just beginning; the implementation will take many years and thus in some areas, improvement is needed:

- frequent reviews and metrics for assessment should be implemented in order to measure the performance and velocity of development teams, quality, and time-to-market of the products and their enhancements
- the Scrum template and a virtual "scrum" task board should be implemented in VSTS. Now distributed teams have in both locations their own task board on the wall.
- stakeholders should give their feedback through product owner and Scrum master, not directly to the development team

- the requirements documents are worse than earlier, an emphasis should be put to defining what makes a sufficient user story and to learning to write them
- collaboration with country units should improve in order to get synergies from developing the backlog items
- PLT roles and responsibilities are not clear to everyone, the way of working should be clarified
- clarification of the roles is needed: for example Scrum master should be more leading the team, more proactive, and know the work items and problems that arise for the current sprint; Product owner should be the silent partner during the sprint and the Scrum master should take the lead.

4.2 Effects on the organization

The implementation of an agile methodology involves high levels of organizational changes and requires crucial executive support to be sustainable and successful (Cohn & Ford 2003, 76). According to Forrester report, changing the behaviour of an IT organization is considered to be demanding and risky (Moore & Barnett 2004, 4). One key characteristics of a self-organizing system is to be able to adapt (Schwaber & Beetle 2002, 117).

Effects on the organization include changes in project and product management and in testing practicalities. On one hand, resistance have been identified; on the other hand, change awareness, personal growth, and some innovative applications of agile practices. The customer involvement is seen more often as an inevitable trend. The organisational structure gets also blames and suggestions for improvement.

Project management responsibilities have changed; earlier project owner had a good control over all projects. Now the responsibility is distributed to each development team. Currently, there is less reporting from projects to project owner and from project owner to management. Project directives are also written in more agile way. If the old way of steering the projects is not changed, the achievements of agile projects are destroyed. Project reporting

in general should be changed to suit better to the agile way: velocity should be reported instead of remaining hours etc. "It is a constant change in small steps. Every day I learn something new."

Ultimately, a project is not agile if its culture does not support agile values. Such a project may regardless benefit from adopting agile practices. (Schuh 2005, 13).

Product management has many challenges. The product backlog for most products has requirements from two different markets and balancing them is difficult. However, backlog is reviewed and prioritised in the sprint planning and everyone has now more understanding of development decisions. The requirements process has changed, they are done shortly before the sprint or during it. The communication about requirements has increased both to development unit and in country units. Special product "virtual team" makes pre-studies and requirements with the product manager for the country unit, resources from different teams are used flexibly in this team. The product managers are not sure of the level of involvement they should have in Scrum team meetings and activities. PLT has now deep product knowledge but a business mindset would be needed: ROI calculations should be used in product backlog prioritisation to ensure sustainable business.

Stakeholders should be active participants in the project - not just people the development team interfaces to (Ambler 2008). They should provide information in a timely manner, make decisions in a timely manner, and actively participate in business-oriented modelling (Ambler 2006). For example product visioning and business planning need active involvement from product marketing in order to deliver the ultimate customer value at the point-of-sale (Highsmith 2004, 2-3).

For testing, resources outside development are needed. Despite the fact that customer support and consultancy have to fulfil their primary obligations first, testing is performed faster today. Sometimes this creates clashes between teams since the need and timing of resources cannot be estimated accurately beforehand. Testers have to respond to needs in short notice and this inflicts

on stress. Management should make some concrete decisions to show that testing is important; more testing resources are needed in development teams. "I could be more proactive and I also should be when working in agile way, the test items are not pushed to me, I pull them to myself"

Allocation of resources in advance used to be rigid since the planning processes were earlier tied to calendar time and not to real time incidents in the markets. When the company moves from year calendar planning to more flexible adaptive planning process these difficulties can be solved (Doz & Kosonen 2008, 147). On the other hand, in agile development projects the need of resources is more or less continuous. People who have variety of roles are torn in several directions.

The organisation structure has been the same and people have been working in the same positions with the same processes for a long time. They are doing their best but a culture of continuous evaluation and improvement is not present. Hence the new way is difficult to assimilate, particularly since the workload is so heavy, the old roles and processes are disturbing and keeping the mindset in waterfall. In an agile team, everybody codes. Currently in the development unit, there are many other roles and people may have fear of their place. Today "T-people" are preferred: they have broad shallow understanding of many issues and deep knowledge in some areas.

To become an agile software developer means moving from a narrowly focused specialist to generalizing specialist (Ambler 2006), who has one or more technical specialities (e.g. Java programming, project management, database administration), has at least a general knowledge of software development, has at least a general knowledge of the business domain in which they work. And who actively seeks to gain new skills in both their existing specialties as well as in other areas, including both technical and domain areas.

On real-world projects, it is sometimes impossible to get an entire team to shift over to an agile value set, much less the customer, stakeholders, or directors who oversee the project. (Schuh 2005, 13.)

One management team worked in an agile way for one month since they had some severe problems. First and second line support have tried out daily meetings for getting better control of customers SLAs (Service Level Agreement). Others see that agile methods could and should be used in more teams and areas, for example meeting conventions could benefit from Scrum time-boxing or customer support or consultancy could prepare user instructions and training material along the sprints or shortly afterwards in more agile way.

Unfortunately perfect communication is impossible. “Managing the incompleteness of communications” is core to mastering agile software development (Cockburn 2007, 1). For example, rather than trying to make a requirements document or design model comprehensive to everyone the document should be sufficient to the purpose and intended audience.

Deeper customer involvement is raised; could customers participate demos with communicator, videoconference, or by some other means. Customers should experience our new agile way of working, the product developers should have constant contact to the real users of the product. This is especially important since in the business chain there might be “knots” which hide information, make partly optimising, or deliberately confuse others (Kettunen 2008, 36). This may result in impossibility to develop the product and the processes.

The learning process is still ongoing, people feel confused of the new roles and responsibilities, there are breaks or “holes” in the communication and dark areas between teams and units. The Norwegian model cannot be directly transferred since the organisation in Norway is different. Questions are raised: How could the productivity of teams be measured, and are distributed teams as (cost-) efficient as presumed? More agility is needed in the entire organisation.

The learning process is ongoing and the implementation will take still some time. “The people in the teams have done this because we have given them the responsibility and the authority. And we have good people!”

Some interviewees brought up an idea to change the organisation to a pure product organisation. All the needed people should be in an agile product team, not spread to different functional teams as today. They could focus better on the product and there would be no unnecessary handovers from team to team to customer. The same product team would be responsible of the product all the way to customers. This might create the community spirit our vice president has been asking for: “Don’t forget that we are **one winning team!**”

In some areas, improvement is however needed:

- clarification of roles and responsibilities in project and product management
- evaluation and consideration whether the organisation could be more product oriented instead of functional oriented
- sales contracts of customer specific features should be reviewed and adjusted to agile development principles; thus both the customer and us could commit to the contract
- customer support could take more advantage of the knowledge gained during the sprints for renewing user instructions and training materials before the delivery
- more flexible delivery of testing resources; consideration of having testing resources inside development teams.

4.3 Effects on the management

When the development teams change the project management and software development processes it creates a domino effect. Collaboration with the development projects’ stakeholders changes and this in turn affects the entire organisation. Furthermore, prevailing management processes have to be evaluated. At least, changes to the following have to be considered:

- line management and general management

- business management and
- product portfolio management.

Some management processes and practices have been changed at the same time as agile implementation, but it has not directly been caused by that; there was a general urge to change these processes. Regardless, business planning in particular needs some development.

Collaboration with management is needed; management initiated the roll-out but otherwise they have not participated or interfered to the implementation. They are supportive, though. Management group gives “the big picture” and directions where to go, and they make decisions to remove obstacles. Management should act in agile manner in order not to be bottlenecks. Top management requires the old way of reporting - but those reports cannot be provided any more since they are not available in agile projects. The reporting chains and manners are somewhat unclear. The reporting processes have changed though the reports are not as detailed as before.

Top management urges the development to work in agile way but there are doubts if they really have understood what that means. For example, there is not enough test resources. The team itself should have test resources to best utilise them. Additionally, the expectation of coding hours per sprint should be decreased in order to be able to test and get the quality better. Furthermore, better customer involvement is desirable in development projects – top management should look into this since it needs planning and dealing with several issues.

Teams take more responsibility to themselves that earlier; the manager does not have to go into details. Manager is more like a coach, trainer, or facilitator in the Scrum teams. Line managers may not be appointing all tasks to some persons; specific activities can come directly from Scrum team, for example testing tasks. In some teams, where employees have not found their roles or they are struggling with underperformance, line management must be aware of this challenge to be able to support people. Extra support is needed to people who are lost and challenged with performance issues.

Agile methods make value visible: people not working very hard will object to agile methods since if someone in an agile environment does not contribute – it will become obvious fast (Heusser 2006, 1). Traditional governance focuses on command and control strategies which strive to manage and direct development project teams in an explicit manner. Lean governance focuses on collaborative strategies that strive to enable and motivate team members implicitly (Ambler & Kroll, 2007).

In some processes, the Corporate rules with non-agile processes. For example, old planning cycles in action planning and making development plan are still in use. “I haven’t noticed bigger changes in any other processes.”, “I think we should make changes to other management processes but I don’t know whether it’s due to agile. The PLT effects should be considered in management.”

The business planning could be more up-to-date with the rest of the organization working with the agile manner. Feedback is collected from customers more rapidly than before because of the more rapid product development. The weak messages should be recognized, thoroughly analyzed and acted upon them. The strategic plans could be reviewed more often but it should be quite a solid plan just the same. Action planning could be done in more agile manner. Strategic agility is not same as vitality in a new growth company which is challenging the old stagnant competitors in the same business area; strategic agility is an ability to think and act in a fresh way (Doz & Kosonen 2008, 30).

PLT and agile methods should reflect on each other. Product road mapping process is basically the same as it has been; product roadmaps are made for the next three years. In waterfall model the idea was to predict the future, plan, and implement it - “but it was a big lie” in business plans and product road mapping. The development actions should be mirrored to customer actions; adapting to the recession and other changes in business situation; the development order could be prioritised and changed according to them. Roadmap reviews should be done more often than currently and reflecting to customer needs in sprints and deliveries.

In different areas 2-3 years can sometimes be a very long time - sometimes short. In big product portfolios, big issues are planned and business strategies considered, in small products rapid changes can be done according to current customer requirements and ROI calculations. Currently, it is more difficult to make the three year roadmap since the flexibility causes changes to original plans. Product roadmaps should be in general level to give a direction, not in detailed level, more of a mindset where we are heading. Detailed plans and roadmaps cannot be shown to customers since they can change rapidly according to emerging requirements and continuous prioritisation.

The agile development may be more productive than the waterfall process, agile enables faster reflection: we can develop new solutions while the iron is hot. Cooperation with customers and sniffing the winds of change is therefore important. When the velocity of the teams and the estimates of backlog items are known and the Scrum team commits to realistic amount of work, estimates and a roadmap for the next year can be done with better predictability than in waterfall model. Otherwise, planning and making external commitments with customers or intertwined teams is not possible (Keith & Cohn 2008, 26).

“Currently the general attitude is agile - if problems arise they are taken more quickly and not postponed.”

Recently the notion of CAS (complex adaptive system) has been adopted from biology and argued to describe a fully agile self-organising team which have some basic guiding practices but not heavy hierarchical organization or rigid management and external control (Augustine & Woodcock 2003, 4). When perceiving their increasingly chaotic environment characterised by rapid changes and increasing amount of information, the teams find change as an opportunity to survive (Maula 2006, 176).

Suggestions for consideration:

- line managers should support people in order to everyone finding their place, in particular clarify the roles, responsibilities, and reporting practices in PLT

- management reporting processes should be evaluated and fine-tuned
- new metrics for calculating the productivity and quality should be developed and implemented. They could include: velocity of the development teams, quantity and throughput time from idea to delivery, defect inflow, percentage of tests passed.
- product roadmaps should be communicated to entire organisation
- business proposals and contracts should be reviewed to see whether they are aligned with the agile development or changes are needed.

4.4 Findings from the study

To be truly agile, a project and the individuals who oversee and regularly interact with it must subscribe to the values and principles of agile. That is, to be truly agile, a project culture must trust its people, foster communication and collaboration, value the ability to respond to change, and never forget that the goal of the project is to deliver software that is useful to the customer (Schuh 2005, 13). If the practices are disconnected from the philosophy (the Agile Manifesto) the result does not work (Heusser 2006, 1). For example, expectations of exact plans and schedules, hard deadlines, expectations of full delivery etc. do not belong to agile mindset.

Here again some remarks from the literature, which seem to be closely aligned to topics discussed in the empiric study. Additionally, some serious words of warning of leaving the implementation in halfway and not impose on continual improvements.

Major suggestions for improvements are listed in the earlier chapters:

- chapter 3.4 Roll-out page 44
- chapter 4.1 Achievements page 50
- chapter 4.2 Effects on organisation page 55
- chapter 4.3 Effects on management page 58.

Basically, they are meant as an introduction for discussions. The individual employees, teams, and managers should consider the suggestions whether any actions are needed or not. "Your manager, colleague, or customer is not

the opponent you should kill – the problems preventing delivery are!” (Cockburn 2007, 430) Remember – **there is only us**. When confronting problems: face them, discuss them openly, and solve them together.

Testing and quality attracted great attention in many of the interview answers. Both test resources and test experience and competence were considered to be insufficient. Additionally the allocation of resources is sometimes seen inflexible. Resources should be available flexibly during each sprint as soon as required; waiting for them either creates slack time between implementing new sprint items or forces the developer to resume to previous item when errors found. Various suggestions for improvements arose; each sprint should have dedicated testes, customer support persons or developers could take turns to test in the sprints, specific testers should be hired to development teams etc.

The literature recommends automated regression testing (Heusser 2006, 1 and Cockburn 2002, xxii) as well as test-driven-development and automated unit testing (Schuh 2005, 105, 116). The time and money invested in them will pay back in reduced time spent on identifying and correcting defects, increase team’s velocity in “quick” changes, and improve quality (Schuh 2005, 108).

Quality in a software product is a complicated matter; it is not only referring to a product without errors, albeit this is highly preferable, but to the correspondence between product features and customer needs as well. In addition, the product should be easy to use, self-instructive and the workflows in the product should comply with the actual customer processes.

In case of prolific amount of defects with substantial effects on cost or customer satisfaction, a root cause analysis of the defects may be in order. Ninna Järvinen is preparing a Master’s thesis on this burning topic.

We also have to respect and evaluate the current positive characteristics of our organisation: processes, methods, and tools in order to avoid abandoning them while adopting new ones. George Santayana’s statement: “Those who cannot remember past are condemned to repeat it!” is only half true (cited in Boehm, 2007, 697). The past includes successful histories too. If y we have

not been made aware of them, we are often condemned not to repeat their successes.

Failures have similarities: massive change without experiment; expecting to have everything: all features, on time, on schedule; without a belief of the core philosophies (responding to change and not planning everything in front). Alternatively: experiment on existing projects; start with automated testing; make decisions that conform to the agile manifesto - start with small incremental changes that support these values. (Heusser 2006, 3.)

Implemented poorly, agile methods can leave an organization in worse shape than when it started with all the same problems! Implemented well agile methods provide working software quickly, give the customer the ability to change his mind, and provide insight into the actual progress of the project and thereby realistic schedules. (Heusser 2006, 4.)

5 CONCLUSIONS

In this study, agile project management and software development methodologies were studied based on previous studies and books, articles, and stories from various agile methodologies' practitioners. An empiric study of the transition from traditional processes to more agile way of working was conducted by observing one unit's endeavours and by interviewing diverse stakeholders of the unit.

More and more teams and organisations are following agile approach. The reasons behind implementing agile methods include compelling problems in performance, efficiency, cost-effectiveness, and competitiveness. For example time-to-market or time-to-react is too slow to utilise suddenly opening market windows; organisation is too complex and it creates redundant work; rigid decision processes create bottlenecks; competitors are more agile. Findings from studies give promising results: an iterative approach to software delivery provides ability to make adjustment and correction with minimal time for rework, development teams can reach a state of hyper-activity, which means

ultimate productivity, and even distributed and offshore teams can gain from agile processes.

In some cases, an organisation adopts agile methods because these have become fashionable and “all the others” are already using them. Practitioners warn of such reason, implementing agile to be fashionable does not change things for better, commitment to using the methods, adopting suitable practises wisely and appropriately for each project, and continuous reflection and improvement does. Adopting agile for its own sake, for instance, misses the point of what it can do. It should be implemented to change the organization. If agile is not surfacing previously hidden organizational dysfunction, helping teams meet previously missed deadlines, or improving the company’s culture, then the organization is likely to use a new set of processes to do the same old tasks. Thus, the final goal should always be the demonstrable improvement of processes and results, not adoption alone.

The way how the agile methods are introduced to an organization is important in order to guarantee that change will be firmly established and have expected positive outcome. Since agile development is a method of building software by empowering and trusting people, acknowledging change as a norm, and promoting constant feedback the culture of the organisation will inevitably change. This evolving new culture may raise confusion, ignorance, and resistance which have to be appropriately handled by offering training, sharing information, providing opportunities to participate, and answering all probing questions patiently. A centralised roll-out team or roll-out responsible person should be nominated although the ultimate responsibility of implementation remains in each team. The roll-out responsible could coach the teams, arrange trainings, and monitor and report the progress of implementation.

Although agile methods are basically implemented in development teams, all the stakeholders that is everyone interacting with those teams will perceive the changed practices and have to change their behaviour, too. Everyone has to know the roles and responsibilities in the new organisation and process otherwise the confusion will curtail the achievements concerning performance and efficiency.

Agile methods change reporting conventions, since projects are not planned up-front in detailed level, the status and progress of projects is not obvious unless measurement practices are changed at the same time. Creating appropriate metrics to provide real evidence of the improvements in cost, scheduling, and quality is needed to be able to evaluate the productivity and efficiency of the organisation. Additionally, metrics for measuring the product quality and economical and business value have to be considered.

The implementation process in Business Unit xxxxx can be examined by comparing it to the eight step process of successful change by John Kotter (Kotter & Rathgeber 2006, 130-131):

1. Create a sense of urgency
Difficulties with time-to-market, time-to-react, productivity, and efficiency were generally identified as compelling problems. They were stated as reasons for the change but there was practically no need to convince the organization since they were known problems.
2. Pull together the guiding team
The vice president promoted agile methods in his info letters and speeches, quality & process manager prepared a presentation to be spread to the organization and he with a (Scrum) project manager with previous experience of agile development planned the implementation. The last mentioned person also prepared and held trainings for different teams.
3. Develop the change vision and strategy
Norway's positive achievements were seen as the goal. Presentations described methods and practices which were the starting point in the new way of working.
4. Communicate for understanding and buy in
Presentations were sent out to managers and product line teams. Info letters to all personnel by the vice president included short promotion of new methods. Trainings were held in development teams and to development projects' stakeholders such as product managers. Testers were invited to trainings, too.
5. Empower other to act

First three teams were encouraged to try agile practices and to reflect on their experiences and make improvements. Later on other teams were invited to join along.

6. Produce short-term wins

The pilot project achievements were published and acknowledged for example in wikis and blogs.

7. Don't let up

Starting the new agile methods has created confusion, some resistance and aptitude for waterfall exists. Further actions are needed.

8. Create new culture

Changing the organizational behaviour takes time to embed but being agile means continuous changes to improve the process. Thus the new culture is created when teams relentlessly seek for better.

“It is not about agile, it's about success” (Pollice 2009); we did not implement agile methodology because it is trendy or because everyone else is doing it. We implemented agile to improve our processes, to be more competitive, and to succeed in our business. We have just started the way but with committed people and some experts the processes will mature and we will evolve into an agile and successful organisation.

In this study, a shallow presentation of chosen topics, agile project management versus traditional project management and agile software development versus traditional waterfall process is provided. Alternative approach could have been to penetrate deep into one area. Nevertheless, this was not considered appropriate since the objective was to study the transition and its consequences in general level.

Reading of theory as a method to familiarise to the topic proved to be exhausting. Significant amount of academic studies exist and even more extensive amount of practitioners of some agile methodology have published information in means of books, articles, internet blogs etc. Selecting reliable and essential references has been difficult. Regardless, some informal sources for example blogs, have also been used if they are aligned with the observations of the case or the interviews. Additionally, some internal sources

have been used in presenting the case and the new processes (product management and product lifecycle).

Agile methodologies are constantly evolving due to the increasing popularity. The underlying principles of agile promote adjusting the methods to current situation and improving them by frequent reflection. Publishing articles and blogs in the internet and in the company's intranet are ways of spreading the good practices and cautionary examples in the agile development community for others to learn and I therefore considered them as respectable source for the study.

The interviews as qualitative research method for this case study proved to be a reasonable research method. The subjects for the interview were carefully chosen to represent various and diverse management levels, processes, functional teams, and countries. None of the invited interviewees refused to answer thus the original plan was realised. From management there were representatives from top and middle management levels and a Scrum master which represents the project management level. There were representatives from product and project management, development, sales, delivery and customer support processes from all four countries. They acted as instance of different functional team as well.

After the interviews an insufficiency of data was identified concerning project management process and practices. Although, the process is significantly different from before and decision process, project phases, document templates, checklists, and examples of the process do not exist any more, the interviewees only brought this issue up few times. Either the change has been so simple and welcome that they did not regard it as something that should be mentioned or they have not understood the magnitude of the change yet.

Since the interviews in this study did not give much insight to the former project managers changed job description and responsibilities while the change of project management from traditional to Scrum is however substantial. A qualitative study of this could be of interest. The study could include discussion of project documentation, reporting and measuring.

Another suggestion for further study is to discuss agile management practices. Many of the interviewees were anxious to see more agility in management teams and processes but the changes were vague and scarce until far. Study question could include, how top management can reflect to changing organisational behaviour and external environment with more agility. Furthermore, a scientifically collected and measured evidence of the business and economical benefits of agile methods would be interesting.

I want to conclude this study with the famous words of Henry Ford:

“Anyone who stops learning is old, whether at 20 or 80. Anyone who keeps learning stays young. The greatest thing in life is to keep your mind young” (cited in Jones 2007, 1). Keeping this in mind helps us to embrace the change, although it is sometimes hard, since changes are only possible by learning and that keeps us young.

REFERENCES

- Abrahamsson, P. 2007. Speeding up embedded software development. Application of agile processes in complex development projects. ITEA: AGILE. 3-6, Newsletter #1/2007. Accessed 21.4.2009. http://www.agile-itea.org/public/deliverables/D.6.4.5_ITEA-AGILE-Newsletter1-2007.pdf
- Abrahamsson, P. 2008. End of Agile. Lecture in OO Days at TUT 25.11.2008. VTT: The Good, Bad and the Ugly of Agile Methods. Tampere. <http://www.cs.tut.fi/tapahtumat/olio2008/esitykset/abrahamsson.pdf>
- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. 2002. Agile Software Development methods. Review and analysis. VTT Publications 478. Accessed 20.4.2009. <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>
- Ambler, S. 2006. Crazy Talk: Examining Why Agile Software Development Works. Accessed 17.4.2009. http://download.developers.sun.com.cn/event/javaone/2006/TS-3987/TS-3987.pdf?bcsi_scan_4E3E68EE5FCEA898=0&bcsi_scan_filename=TS-3987.pdf
- Ambler, S. 2008. Agile project planning tips. Accessed 29.4.2009. <http://www.ambysoft.com/essays/agileProjectPlanning.html>
- Ambler, S. W. & Kroll, P. 2007a. Best practices for lean development governance -- Part 1: Principles and organization. June 2007. Accessed 3.9.2008. <http://www.ibm.com/developerworks/rational/library/jun07/kroll/index.html>
- Ambler, S. W. & Kroll, P. 2007b. Best practices for lean development governance -- Part II: Processes and measures. July 2007. Accessed 3.9.2008. http://www.ibm.com/developerworks/rational/library/jul07/kroll_ambler/index.html
- Ambler, S. W. & Kroll, P. 2007c. Best practices for lean development governance -- Part III: Roles and policies. August 2007. Accessed 3.9.2008. http://www.ibm.com/developerworks/rational/library/aug07/ambler_kroll/index.html
- Anderson, D. J. 2004. Agile Management for Software Engineering. Applying the Theory of Constraints for Business Results. Upper Saddle River: Prentice Hall.
- Anderson, D. M. 1997. Agile Product Development For Mass Customization, JIT, Build-to-Order, and Agile Manufacturing. Accessed 23.9.2008. http://www.scpdnet.org/papers/dma-agile/agile_prod_dev.htm
- Anthes, G. 2005. The Future of IT. Computerworld, 27-36, March 7, 2005.
- Augustine, S. 2006. Managing Agile Projects. Upper Saddle River: Prentice Hall PTR.
- Augustine, S. 2007. Transitioning to Agile Project Management. Realizing Customer Value through Throughput, Teamwork and Leadership. Lecture in Agile 2007 conference. Accessed 15.5.2009. http://www.agile2007.org/downloads/handouts/Augustine_474.pdf

Augustine, S. & Woodcock, S. 2003, Agile Project Management. Accessed 15.4.2008.

<http://www.ccpace.com/Resources/documents/AgileProjectManagement.pdf>

Balduino, R. 2007. Introduction to OpenUP. Accessed 3.9.2008.

<http://www.eclipse.org/epf/general/OpenUP.pdf>

Beck, K. & Andres, C. 2004. Extreme Programming Explained. Embrace Change. 2nd edition. US: Addison-Wesley Professional.

Boehm, B. 2007. A View of 20th and 21st Century Software Engineering. In Software Engineering. Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research. Edited by Selby, R. W. 2007. Computer Society. New Jersey. 679-729.

Braithwaite-Lee, R. 2002. Agile methods in a product development setting. Accessed 21.5.2008. <http://www.braithwaite-lee.com>

Burke, E. 2008. Eric Burke talks about moving into more of an enterprise adoption. Video interview from Agile 2008 Conference. Toronto.

<http://www.agilejournal.com/media-center/videos?task=searchvideos&searchkey=Agile>

Cockburn, A. 2002. Agile Software Development. Boston: Addison-Wesley.

Cockburn, A. 2007. Agile Software Development. The Cooperative Game. Second edition. Boston: Addison-Wesley.

Cockburn, A. 2008. Effective Software Development In The 21st Century: The New Face Of Software Engineering. Lecture in OO Days at TUT 25.11.2008. Tampere.

<http://www.cs.tut.fi/tapahtumat/olio2008/esitykset/cockburn.pdf>

Cohn, M. 2005. A Visual Introduction to Scrum. Accessed 6.3.2009.

<http://www.mountaingoatsoftware.com/scrum>

Cohn, M. 2006. Metrics You Can Bet On. Better Software. August 2006. Better software magazine. © 2009 StickyMinds.com

<http://www.stickyminds.com/BetterSoftware/magazine.asp> Accessed 10.5.2009.

<http://www.mountaingoatsoftware.com/articles/24-metrics-you-can-bet-on>

Cohn, M. 2008. Patterns of Agile Adoption. Agile Journal. January 2008. Agile journal. <http://www.agilejournal.com/> Accessed 10.5.2009.

<http://www.mountaingoatsoftware.com/articles/38-patterns-of-agile-adoption>

Cohn, M. & Ford, D. 2003. Introducing an Agile process to an organization. Computer. June 2003, 74-78.

Coplien, J. 2008. To Those who were Agile before Agile was cool. Lecture in OO Days at TUT 26.11.2008. Tampere.

<http://www.cs.tut.fi/tapahtumat/olio2008/esitykset/coplien.pdf>

Cottmeyer, M. 2009. Leading Agile: Why Agile will Fail. A blog on Agile leadership and Project Management. Accessed 11.5.2009.

<http://www.leadingagile.com/2009/01/why-agile-will-fail.html>

Doz, Y. & Kosonen, M. 2008. Nopea strategia. Miten strateginen ketteryys auttaa pysymään kilpailun kärjessä. Suomentanut Mauri Laukkanen. Helsinki: Talentum.

Foot, M. & Hook, C. 1999. Introducing Human Resource Management. 2nd edition. Harlow, England: Addison Wesley Longman.

Fowler, M. 2002, The XP 2002 Conference. Accessed 5.3.2009.
<http://martinfowler.com/articles/xp2002.html>

Fritzsche, M. & Keil, P., 2007. Agile Methods and CMMI: Compatibility or Conflict?, e- Informatica Software Engineering Journal, Volume 1, Issue 1, 2007. 9-26.

Frye, C. 2007. Making agile software development work for distributed teams. Software quality news. Accessed 20.5.2008.
http://searchsoftwarequality.techtarget.com/news/article/0,289142,sid92_gci1277064,00.html

Glazer, H., Dalton, J., Anderson, D.J., Konrad, M. & Shrum S. 2008. AMMI and Agile: Why Not Embrace Both! Technical Note CMU/SEI-2008-TN-003 Software Engineering Institute. Carnegie Mellon University. Accessed 18.5.2009.
<http://www.sei.cmu.edu/publications/documents/08.reports/08tn003.html>

Gustafsson, B. 2008. OpenUP –The Best of Two Worlds. Methods and Tools. Accessed 3.9.2008. <http://www.methodsandtools.com/archive/archive.php?id=69>

Hass, K. 2007. The Blending of Traditional and Agile Project Management. PM World Today. May 2007. Vol IX, Issue V. Accessed 2.9.2008.
<http://www.pmforum.org/library/tips/2007/PDFs/Hass-5-07.pdf>

Heusser, M. 2006. Perils and Pitfalls of Agile Adoption. Accessed 16.4.2009.
<http://www.informit.com/articles/article.aspx?p=441115>

Highsmith, J. 2004. Agile Project Management: Creating Innovative Products. Addison-Wesley. Boston: Pearson Education Inc.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2004. Tutki ja kirjoita. 10 p., osin uud. laitos. Helsinki: Tammi.

IPMA. International Project management association. This is IPMA. Accessed 22.4.2009. <http://www.ipma.ch/Pages/default.aspx>

Jacobson, I. 2008. Developers are too fashionable. Interview of Ivar Jacobson.
<http://www.builderau.com.au/strategy/developmentprocess/soa/lvar-Jacobson-Developers-are-too-fashionable/0,339028278,339291697,00.htm>

Jens. 2007. Can distributed teams be agile. IntelliJens blog. Accessed 20.5.2008.
<http://intellijens.se/2007/03/can-distributed-teams-be-agile.html>

Jones, T. 2007. Henry Ford Community College: Learning Lessons from the Past. Accessed 20.4.2009. <http://www.certmag.com/read.php?in=2838>

Kainulainen, A. 2008. Agile-menetelmät. Master's thesis. Jyväskylä university of applied sciences, Information technology.

Kalermo, J. & Rissanen, J. 2002. Agile software development in theory and practice. Master's thesis in Information systems science. University of Jyväskylä. Department of computer science and information systems. Jyväskylä.

Keith, C., & Cohn, M. 2008. How to fail with agile. 20 tips to help you avoid success. Better Software. July/August 2008, 24-28. Accessed 10.5.2009
www.stickyMinds.com

Kettunen, J. 2008. Uudistu ketterästi: kehitä, osta ja varasta. Helsinki: Talentum.

Kniberg, H. 2007. Scrum and XP from the Trenches. USA: InfoQ.

Kussmaul, C. 2004. Agile Product Development: Lessons from Industry. Education that works: The NCIIA 8th annual meeting. March 18-20, 2004. Accessed 2.9.2008
http://www.nciia.org/conf_04/proceedings_04/htmldocs/papers/kussmaul.pdf

Kroll, P. 2007. OpenUP In a Nutshell. Rational Edge, Sep. 2007. Accessed 3.9.2008.
<http://www.ibm.com/developerworks/rational/library/sep07/kroll/index.html>

Kvale, S. 1996. InterViews. An Introduction to Qualitative Research Interviewing. California: SAGE Publications.

Lehtimäki, T. 2006. Ohjelmistoprojektit käytännössä. Helsinki: readme.fi.

Loeser, A. 2006. Project Management and Scrum – a Side by Side Comparison. Accessed 10.5.2008.
<http://research.ittoolbox.com/white-papers/itmgmt/pm/project-management-and-scrum-a-side-by-side-comparison-2462>

Maula, M. 2006. Organisations as Learning systems. 'Living composition' as an Enabling Infrastructure. Oxford: Elsevier.

Meredith, J. & Shafer, S. 2002. Operations Management for MBAs. Second Edition. Denvers, MA: John Wiley & Sons, Inc.

Moore, G. A. 2005. Crossing the chasm. Second edition. Chichester, West Sussex: Capstone.

Moore, S. & Barnett, L. 2004. Offshore Outsourcing And Agile Development. Forrester Research Inc. Accessed 10.5.2008.
<http://www.forrester.com/Research/Document/Excerpt/0,7211,34978,00.html>

MSDN. Team System home. Team system developer center. Accessed 1.5.2009.
<http://msdn.microsoft.com/en-us/teamsystem/default.aspx>

Palomäki, P. 2008. Ketterä ohjelmistokehitys – globaalia vai paikallista. Accessed 10.10.2008. http://www-05.ibm.com/fi/ibm/ibmgives/university/pdf/Kettera080924_Case_F-Secure.pdf

Pikkarainen, M. 2008. Towards a Framework for Improving Software Development Process Mediated with CMMI Goals and Agile Practices. Academic dissertation. the University of Oulu, Department of Information Processing Science. VTT publications 695.

Pikkarainen, M., Salo, O. & Still, J. 2005. Deploying Agile Practices in Organizations: A Case Study. In Software Process improvement. 12th European Conference, EuroSPI 2005, Budapest, Hungary. Editors Richardson, I. Abrahamsson, P. & Messnarz, R. Springer. 16-27. Accessed 19.4.2009.

http://books.google.fi/books?id=COpuwuxNQMcC&pg=PA16&lpg=PA16&dq=Deploying+Agile+Practices+in+Organizations+Pikkarainen&source=bl&ots=tmP-MjNWfh&sig=8Ejy_RKT7dwochCpIKXchYM24R4&hl=fi&ei=78PtSZCgHlyX_AaKwoDsDg&sa=X&oi=book_result&ct=result&resnum=1

PMBOK Guide. 2004. A Guide to the Project Management Body of Knowledge. Third edition. Pennsylvania, USA: Project Management Institute.

PMI. Project Management Institute. Accessed 22.4.2009.

<http://www.pmi.org/Pages/default.aspx>

Schuh, P. 2005. Integrating agile development in the real world. Hingham, Massachusetts, USA: Charles River Media, inc.

Schwaber, K. 2007. The Enterprise and Scrum. Redmond, Washington, USA: Microsoft Press.

Schwaber, K. & Beedle, M. 2002. Agile Software Development with Scrum. Upper Saddle River, New Jersey: Prentice Hall.

Scumniotales, J. 2009. Agile Product Development. Why incremental development is better – An ROI perspective. Accessed 30.4.2009. <http://scrumone.typepad.com/>

Stark, J. 2006. Product lifecycle management. 21st Century Paradigm for Product Realisation. 3rd printing. London: Springer.

Stepanek, G. 2005. Software Project Secrets. Why Software Projects Fail. Berkeley: Apress.

Sutherland, J. 2008. Self-Organization: The Secret Sauce for Improving your Scrum team. 4. September 2008. Accessed 18.5.2009. <http://jeffsutherland.com/scrum/>

Sutherland, J., Viktorov, A. & Blount, J. 2006. Adaptive Engineering of Large Software Projects with Distributed/Outsourced Teams. Accessed 11.9.2008. <http://necsi.org/events/iccs6/papers/ee6637fd0a1f958002d8f242162b.pdf>

Szalvay, V. 2004. An Introduction to Agile Software Development. Accessed 17.4.2009. http://www.danube.com/docs/Intro_to_Agile.pdf

Tadipatri, S. Implementing agile in distributed teams. Presentation at Agile Goa 2007 conference. Accessed 20.5.2008. <http://agileindia.org/agilegoa07/index>

The Agile Manifesto. 2001. Accessed 3.4.2009. <http://agilemanifesto.org/>

W2E. Software development process. Product management process. Process intranet in Corporation xxxxxxxxx. Accessed repeatedly.

West, D. 2009a. Best practices: Software Development Processes. Forrester Research Inc. April 15. 2009. www.forrester.com

West, D. 2009b. Agile Processes Go Lean. Dr. Dobb's Portal. Architecture & Design. 25. April 2009. Accessed 18.5.2009. <http://www.ddj.com/architect/216600161>

APPENDICES

Appendix 1: Definitions

Agile **Agile** can refer to: Agility, Agile management, Agile software development, a family of software development approaches, Agile Manifesto etc.

Agility In business, **agility** means the capability of rapidly and cost efficiently adapting to changes. Recently agility has been applied e.g. in the context of agile software development and agile enterprise.

Agile management

Agile management takes the ideas from Agile software development and applies them to management in general. Of the software-related agile methods, Scrum is usually considered the most non-specific to software. However, Extreme Programming has also been used for managing non-software projects. Agile Management also takes ideas from Lean Manufacturing and general team building methods. Agile work is the most general expression of agile management.

Agile software development

Agile software development is a group of software development methodologies that are based on similar principles. Agile methodologies generally promote a project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering best practices that allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals. Conceptual foundations of this framework are to be found on modern approaches to operations management and analysis such as Lean manufacturing, Soft Systems Methodology, Speech act Theory (Network of conversations approach), and Six Sigma.

Agile Manifesto

The **Agile Manifesto** (<http://agilemanifesto.org/>) is a statement of the principles that underpin agile software development. It was drafted in February 2001 by representatives of various new methodologies such as Extreme Programming, Scrum, DSDM, Adaptive Software Development, Crystal, Feature Driven Development, Pragmatic programming, who met to discuss the need for lighter alternatives to the traditional heavyweight methodologies.

ASP An **Application Service Provider** (ASP) is a business that provides computer-based services to customers over a network. ASP is firm that sells usage of computer programs via internet. An ASP (equipped with all required software, hardware, and trained employees) guarantees trouble-free availability of the application programs on a continuous basis. Customers use the programs they need, for a fixed monthly fee or usage based charges. The data generated by those

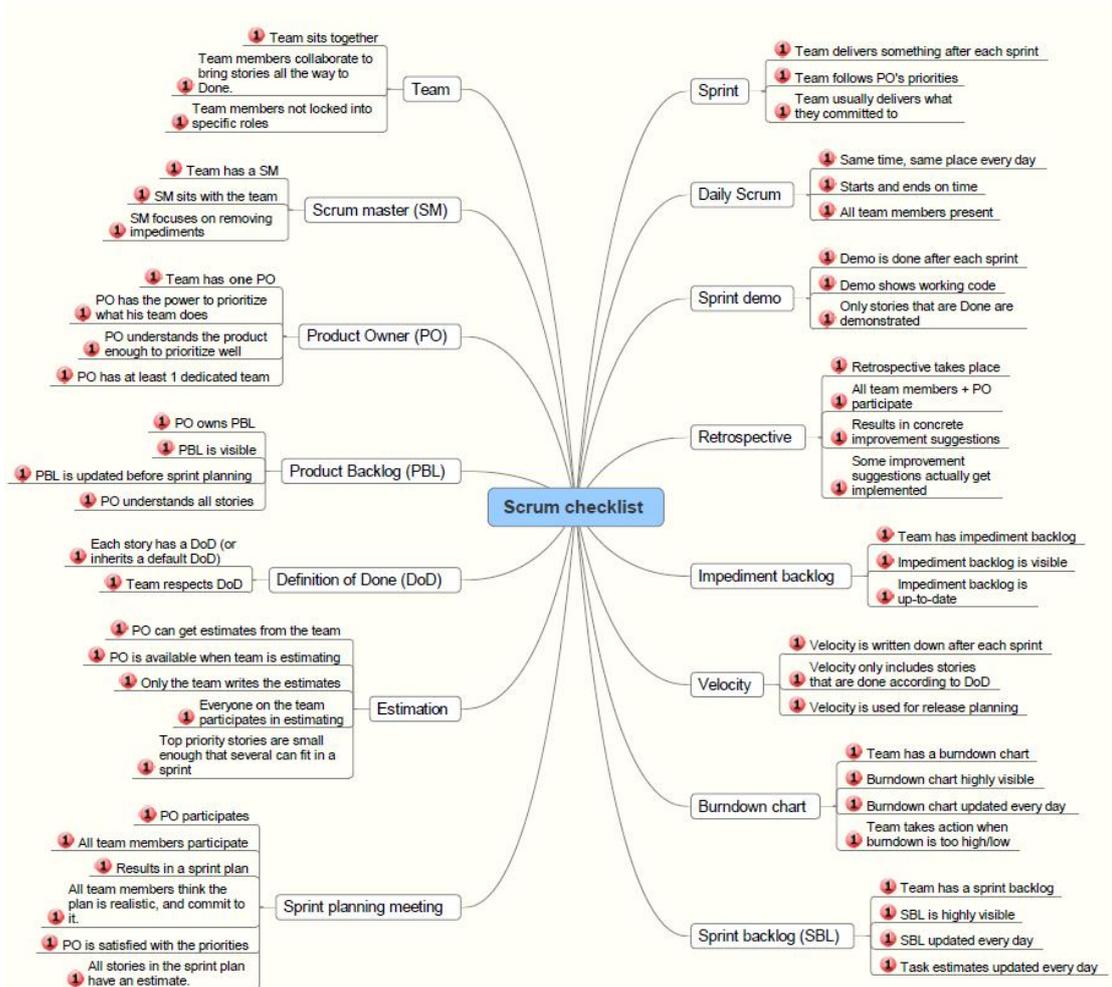
programs can either be stored on the customer's computer or on the disk space rented out by the ASP on its storage devices.

Lean **Lean manufacturing** or **lean production**, which is often known simply as "**Lean**", is a production practice that considers the expenditure of resources for any goal other than the creation of value for the end customer to be wasteful, and thus a target for elimination. In a more basic term, "More value with less work." Lean manufacturing is a generic process management philosophy derived mostly from the Toyota Production System (TPS) and identified as "Lean" only in the 1990s

Waterfall development model

The **waterfall model** is a sequential development process, in which development is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design (validation), Construction, Testing, and Maintenance.

Appendix 2: Scrum checklist by Henrik Kniberg



Appendix 3: Case study

Motivation letter and questions for the interview

Agile implementation

Background

One year ago we in Business Unit xxxxx started to pilot agile methods and practises in our product development. Now we have used these methods for some time in all our development teams and have gained experience.

This questionnaire aims to draw a map of the implementation of Agile methods and practices in Business Unit xxxxx; why we chose to start using agile, how the change process was carried out, what was the roll-out like, what problems have we encountered, how have we solved them.

The theme of the questionnaire is especially to find out the effects of agile implementation on all stakeholders in the organization: how the implementation of agile in software development team influences other teams, processes, management etc.

It is important to get the thoughts and opinions from people in different roles in the organization; business managers, product mgmt, sales, customer support & delivery, product development etc.

Please consider the questions from your own point of view, make some notes and be prepared to the interview in near future. If you send your notes to me in advance, it will be of great help.

BR
Helinä

Questions

Background info

1. What is your role and responsibilities in the organization? How do you relate to product development? How do you relate to product or business management. If you have several roles, please consider the following questions through all of them.

Identifying stakeholders and the effects on them

2. Which, in your opinion, are the stakeholders that are affected when product development starts using agile methods?
3. What agile methods, practices, techniques etc. have you identified to be in use (that have had a direct or indirect effect on your work)? What else should still be implemented?
4. How does the change to agile affect your work? What has changed from before?

Roll-out process

5. How was the roll-out of agile carried out? What could have been done better in the roll-out? What should still be done?
6. Why, in your opinion, did we start to use agile methods? What were the expected benefits and outcome?
7. What has been achieved until far?
8. By which means these achievements have been attained?
9. Have there been any drawbacks or problems? In which areas, teams, processes, etc. in particular?
10. What might be the reason for these problems?
11. How have these problems been solved? Or if not yet - how could they be solved?
12. How could the problems be prevented if another organization would implement agile methods?

Effects on business, product, and other management processes

13. Have you done or is it necessary to make changes to annual business planning, product management roadmapping etc. management processes?

General questions

14. Has the organization and its processes become more agile in general? Or should some changes still be done? What and How?
15. Anything else you wish to point out?

Roles of the interviewed persons

- 1 Vice President of IU Business Unit xxxxx, strategy & business responsibility, Sweden
- 2 Sales and product portfolio mgr, business responsibility, country unit, Finland
- 3 Product manager, customer needs and requirements responsibility., country unit, Finland
- 4 Product development manager, project owner, development unit, Sweden
- 5 Sales manager, team leader, Sweden
- 6 Customer support manager, country unit, Sweden
- 7 Customer support, tester, country unit, Finland
- 8 Delivery and configuration development unit, Sweden
- 9 Scrum master, developer, development unit, Sweden
- 10 Developer, development unit, India
- 11 Development manager, Norway
- 12 Developer, Norway

Appendix 4: Case study: Interview answers

Removed from the electronic version.