NORGES HANDELSHØYSKOLE

Bergen, 09.06.2009

NHH's Portal Project

– a study on IT project management from both a managerial and technical perspective

Stian Clementsen

Advisor: Tore B. Holmesland

Master Thesis

MSc. Business Analysis and Performance Management

NORGES HANDELSHØYSKOLE

The Norwegian School of Economics and Business Administration

Abstract

This thesis attempts to acquire insight in IT project management both from a technical and a managerial perspective. It studies the technical methods of agile software development, which are popular in software development, but also look at the need of keeping a PSO perspective, i.e. being aware of people and organization development, as well as system development. It is a case study on the Portal Project at the Norwegian School of Economics and Business Administration (NHH), an ongoing project which was initiated in 2008 and aims to deliver a new operation portal integrating all electronic tools and information for students, researchers, and administrative staff at the school. The project is applying a light version of Scrum development, while maintaining a PSO perspective. Yet it also raises a question whether using milestones should be combined with Scrum would improve the PSO perspective even more without interfering with Scrum's efficient development practices.

Preface

I am about to finish my five years of business studies at the Norwegian School of Economics and Business Administration (NHH) in Bergen, Norway, and this thesis will complete my Master of Science degree in Business Analysis and Performance Management. The thesis is worth 30 ECTS and has been worked upon from January to June 2009.

I find information technology interesting because it is changing so fast, and introduces us to new ways of doing thing both for individuals and organizations, and especially the ways we can communicate and work together across both borders and time zones. It is through IT projects that organizations develop or acquire new technology and systems, and these projects are often challenging simply because technology changes, and also because they require management different from other kinds of projects.

Writing the master thesis and working on a topic like this for almost half a year has both been exciting and challenging. I started out not knowing much more than that I wanted to write about business and IT, and preferably do a case study. And now I feel this is exactly what I have been allowed to do. The portal project at NHH is indeed an IT project, and it has allowed me to get insights which I could relate both to technical and managerial IT project management theories. After this study, there is no doubt that I would still like to be involved in IT projects myself – now when student life ends and work life is soon to start.

I would like to thank my advisor Tore Holmesland, not only for his efforts in counseling along the way and support while I was writing, but also for his inspirational attitude which greatly motivates learning more about IT projects. I am also very grateful for having been granted access to the Portal Project, and for everyone at NHH and Bouvet who made themselves available for interviews in order to talk to me about their experiences. My family deserves many thanks for believing in me, letting me make my own decisions and standing on my own legs, but always being there when I need them. Thanks to the Phusapan and Castore family for accommodating me in Bangkok which allowed me to learn more about the Thai and Asian life and culture. Also, many thanks go to Panida for cheering me up throughout the writing process.

– Stian Clementsen, Bangkok, June 2009

Table of Contents

ABSTRACT **2**

PREFACE **3**

TABLE OF CONTENTS **4**

TABLE OF FIGURES **7**

1. INTRODUCTION **8**

2. PROJECT DESCRIPTION **12**

2.1 INTRODUCTION 12

2.2 PURPOSE OF THE PROJECT 13

2.3 PROJECT DEVELOPMENT METHOD 14

2.4 PROJECT ORGANIZATION 14

2.4.1 Project Managers 15

2.4.2 The Content Part 15

2.4.3 The Portal Part 16

2.4.4 The Process Part 16

2.4.5 The Steering Group 17

2.4.6 The Project Support Team 17

2.4.7 Programmers/Developers 17

2.4.8 Graphical and Interaction Design 17

2.4.9 KPMG 18

2.4.10 Inspera 18

2.5 CURRENT STATUS OF THE PROJECT 19

3. THEORY AND MODEL PRESENTATION **20**

3.1 PROJECT AND PROJECT MANAGEMENT DEFINITIONS 20

3.1.1 Projects 20

3.1.2 Project Management 21

3.2 WATERFALL SOFTWARE DEVELOPMENT 21

3.3 AGILE SOFTWARE DEVELOPMENT 23

3.3.1 The Background for Agile Software Development 23

3.3.2 What is Agile Software Development? ... 24

3.3.3 An Overview of Some Agile Methods ... 24

3.3.4 The Agile Manifesto ... 25

3.3.5 Principles behind the Agile Manifesto ... 27

3.3.6 Criticism of Agile Software Development ... 28

3.4 SCRUM ... 28

3.4.1 Scrum Methodology ... 29

3.4.2 Scrum Roles ... 29

3.4.3 The Pregame and Planning ... 30

3.4.4 The Game and the Sprints ... 30

3.4.5 The Postgame and Closure ... 32

3.4.6 The Backlogs ... 32

3.4.7 The Burndown Chart ... 33

3.5 THE PSO PERSPECTIVE ... 34

3.5.1 The Background for a PSO Perspective ... 34

3.5.2 What is a PSO Perspective? ... 35

3.6 AGILE SOFTWARE DEVELOPMENT VS. PSO ... 37

3.7 WHY ARE IT PROJECTS LATE AND DELAYED? ... 37

4. METHODS AND TECHNIQUES ... 40

4.1 QUALITATIVE METHOD AND CASE STUDY ... 40

4.2 INTERVIEWS ... 41

4.3 VALIDITY AND RELIABILITY ... 42

4.4 ACCESS TO THE PROJECT ORGANIZATION ... 43

5. ANALYSIS AND THOUGHTS ON THE PORTAL PROJECT ... 45

5.1 PROPOSITIONS ... 45

5.2 SCRUM LIGHT ... 45

5.3 COMPOSITION OF PROJECT MEMBERS ... 48

5.4 COLLABORATION ... 49

5.5 HANDLING TASKS IN THE PROJECT ... 50

5.5.1 Entering Tasks in the Backlogs ... 50

5.5.2 Prioritization with Post-It Notes 51

5.5.3 Short-Term Planning 52

5.6 Measuring Project Progress 53

5.6.1 Storing Information on Progress 54

5.6.2 Retrieval of Progress Information 54

5.6.3 Transparency Issues 55

5.6.4 Missing Milestones 56

5.7 Making Decisions in the Project 58

5.8 Degree of Agility and Flexibility 60

5.9 Focusing on the Budget 62

5.10 Applying a PSO Perspective 63

5.10.1 Informing and Creating Awareness 63

5.10.2 Including NHH People in the Project 64

5.10.3 Offering Trainings to the Users 65

5.11 Resistance to Change 66

5.11.1 Resistance Within the Project Organization 66

5.11.2 Resistance From the New Users 67

5.12 NHH as an Academic and Public Institution 68

6. CONCLUSIONS 70

6.1 Answers to Research Questions 70

6.2 Justifying Propositions 72

7. SUGGESTIONS 75

BIBLIOGRAPHY 76

APPENDICES 80

Appendix I: Interview with Project Manager at NHH 80

Appendix II: Interview with Technical Sub-Project Manager at NHH 81

Appendix III: Interview with Content Sub-Project Manager at NHH 82

Appendix IV: Interview with Developers/Programmers at Bouvet ASA 83

Appendix V: Interview with a member of Project Support Team 84

Table of Figures

Figure 1: Illustration of the Portal Project Organization (Source: Interviews) 18

Figure 2: Waterfall Development Process (Royce 1970) 22

Figure 3: Scrum Methodology (Schwaber 1995) 29

Figure 4: Scrum Sprint Model (WikiPedia 2009c)..31

Figure 5: Sample Scrum Burndown Chart (Schwaber 2007)...34

Figure 6: PSO Perspective (Andersen et al. 2004) 36

Figure 7: pSo Perspective (not PSO) 36

Figure 8: Combining Scrum, Milestones, and PSO...58

1. Introduction

I consider myself a business student with considerable interest in information and communication technology. In order to merge these two, I believe a study on managing IT projects would be a very interesting topic to go deeper into, and thus is the reason for my choice of theme for my master thesis.

Most people use different IT systems and solutions throughout their day. Whether it is using applications at work, writing messages on your cell phone, or shopping online from home, you will realize that much of your time is spent interacting with an IT system in one way or another. From a business perspective, IT has improved processes by managing accounting, human resources, project planning, budgeting, forecasting, order handling, etc. The systems become more and more complex, and there seems to be no limits on what IT can do for us, or what new opportunities that will evolve in the future. But – and this is a big but – a decision to develop a new IT system does not mean that you will automatically get all these benefits. You need to know what you want from you system and if the changes it provides for you is worth the investment.

You need to make sure the organization is prepared for changes in both structure and processes. Most importantly the people must be trained to understand and be able to take advantage of the new solutions offered by the system. And if this does not already seem like a lot of work, managing an IT project is not like managing any other project, nor is administering the people producing the deliverables (engineers writing code). Although adding manpower to a construction project might help you reach your deadline for finishing a new building; doing the same with software engineers will most likely delay the project even more (Brooks 1971). This issue and many others distinguishes IT projects from traditional projects.

I know many new technological inventions are yet to come. From past experience through work places, schools, and other organizations, I know there are still much, much more IT can help us with when we know how, and the right way of acquiring it. This excites me, and I believe and hope that I will be able to initiate or participate in such projects in the future. Yet there are many challenges in IT projects. We know this by reading papers and hearing about lots of failing IT projects that are either delayed, exceeding budget, or both.

I believe a broad understanding of many fields is necessary in order to manage IT projects. It is not enough to be an expert on management (if there is such a thing) or an expert on IT alone. There are too many consequences of implementing IT systems, and too much that are afflicted by the changes. First of all, initiating an IT project is an investment, thus one has a financial aspects of return on investment. Such a return can for instance be cost savings by automation, or income generating by offering new services. If we have cost savings there will have to be structural changes, because the IT system cannot offer any improvements if we do the same things the same way after the implementation as we did before (Dahl and Fuglseth 2007).

We need to know and understand change management. If our system will offer new services, we must know if these services are worth anything to our customers. We need to understand marketing and sales. We need to estimate cash flows, market- and competitors' reactions. What can the system itself do? Are there any limits on coding, memory, processing power, bandwidth, etc? Maybe some things can actually prove impossible. We need to have technical insights. We need to understand how the software engineers develop the code, and cooperate with each other. There is so much that can be affected by an IT system, that it is so easy to forget something, which, because it was not taken into consideration, will lead to delays, exceeding budgets, or ultimately complete failures. Murphy's Law is very relevant: “Anything that can go wrong, will go wrong.”

How can we overcome all these challenges? Does the project manager need a master degree in business, management, technology, psychology, and sociology? All these fields prove important for IT projects, yet a project manager simply cannot know them all in depth. My personal impression is that business students and engineering students learn two different kinds of project management. Project management literature written by people with business/management background does not take into consideration the technical challenges in the engineer literature on project management; likewise engineering literature does not take into consideration the challenges in the business/management literature. Thus in my thesis I will do my best to acquire a better knowledge of both fields, and see if some of the models can me merged. Winter et al. (2006) do also recognize that there has come a new class of projects that differ from the traditional engineering view of projects. They call these projects business projects – and IT projects as well as organizational change projects, come under this category.

In business/management literature I have found the People System Organization (PSO) method to be of high interest in project management. And in technical literature the Agile Software Development (ASD) methods seem to be well acknowledged approaches. Both project management methods seem to be very reasonable and good solutions if read separately, but if read together, one can see that they are contradicting each other somewhat. Whereas PSO shows us the importance of not just focusing solely on the system, but that the people and the organization development is just as important to be taken into consideration; the ASD methods teach us much of how the software development itself should be managed. The PSO method does not take the ASD methods view on planning into account, and the ASD methods do not take the PSO method's focus on people and organization into account.

I will first introduce and go through the different project management theories, before trying to see if they can be merged somehow. Further I will do my research on an IT project at the Norwegian School of Economics and Business Administration (NHH). NHH has initiated the so-called Portal Project. This project aims to deliver a new desktop solution that will greatly affect the way students, faculty, and other staff use computers at the school. I will try to see if the project is managed from a business or technical perspective, or maybe both. I will try to see in what ways the project is managed in accordance with the different theories. And we will try to see if the challenges they experience are in accordance with the theory and if the theory can explain the obstacles experienced.

It should be noted that the Portal Project is an ongoing project, which started in the latter part of 2008, and which is not scheduled to finish until 2012. Only the first phase will finish by summer 2009. As a consequence of the project not being finished, one cannot look back, evaluate, and realize what could have been done differently or better, as is done in most situations; nor are any evaluation documents available. Yet this makes the research more exciting. I will describe and analyze some parts of the projects and look at it with Agile and PSO glasses and bring conclusions and predictions more than long time before project termination. Doing research before the project is finished will (hopefully) give the project team valuable information to take into consideration for future planning of the project, and for me personally it can be interesting to have a mid-time paper to look back at and compare with the actual evaluations when the project is terminated. This will be interesting because it will allow seeing if the things turned out the way I have predicted based on the theory, whether I was right or wrong. It might be at least as important to know how to predict the future of a project based on what we're doing when we can still change, instead of being able to look back at the project when it is done, and learn what could have been done differently when it is too late to change. This does by no means suggest not doing evaluation after a project, as this important learning part is essential to project work, yet it would give an interesting perspective trying to predict the future of the project before it is done as well.

By first acquiring theoretical knowledge on IT project management, mainly agile software development (and Scrum) as well as the PSO perspective, I will use this knowledge as a basis to better understand and analyze the portal project. The purpose of looking at a live project is to go beyond theory and see what goes on in the real world and how they handle the theoretical frameworks. My main goal with this thesis is to strengthen my knowledge and understanding of IT project management and agile development and get an impression of how a real project manages the challenges in its environment. I will seek to find answers to the following questions:

- *Does the portal project manage to adapt an agile software development methodology?*
- *How does the portal project make agile software development work for them?*
- *How does the portal project handle the need for a PSO perspective?*
- *What works well, and what is more challenging in the project, and why?*

I believe these questions are essential in order to get an understanding of how agile software development works for this particular portal project, and this knowledge can prove to be handy for other IT projects.

2. Project Description

2.1 Introduction

The Norwegian School of Economics and Business Administration uses a considerable amount of different digital resources and information systems in order to handle a variety of processes and information at the institution available for students, researchers, administration, and other employees. However, many of these systems are not fully integrated and it can be hard for the users to keep track of them all, and especially to navigate between them. Thus the need for an operation portal has evolved in the current decade, and a pre-study was conducted in fall 2006 in order to clarify the need for such a portal at NHH.

As a result of the pre-study NHH's IT Department gave a presentation of the already approved plans of the Portal Project in NHH's IT User Forum in spring 2007 (IT User Forum 2007). Back then one expected to initiate a public invitation to tender in fall 2007, and launching the portal already in spring 2007. Both estimates would later prove to be delayed by a year. However after considerable research by the project's core team in spring 2008, NHH was ready to invite to a public tender later in 2008. 11 qualified contestants were approved to participate in the bidding round. It was important to give all contestants an equal chance. Thus everyone was offered the same amount of meetings and time to present their solutions. Many good offers were received, but the core team would evaluate each and every one based on criteria agreed upon before the bidding round. Among the criteria which were important were that the supplier was of a certain size with concrete and relevant project experience. It was not only important what was available in the company, but also what they were willing to offer and contribute with.

Although NHH had never worked with them before, Bouvet ASA, a Scandinavian IT consultancy firm, won the bidding in fall 2008. They proved to be highly competent as well as having done many similar projects earlier. NHH wanted to make sure the best engineers were working on the development. Because of this, most of the portal would be developed in Stavanger where one had the best knowledge on using a Microsoft Office SharePoint Server. NHH is satisfied with the Chief Programmer having considerable experience and a solid background. NHH also found the Scrum development process offered by Bouvet interesting, as it promised fast and iterative deliveries, showing immediate results.

Bouvet started working on the project in late 2008, but is now already getting close to finishing the first phase, by delivering a working portal already before summer 2009. Many people are excited to see what is going to be delivered. The project is quite unique as being one of the largest projects at NHH in recent time, and definitely the largest within the IT Department. For an institution not used to working with projects of this scale, yet with considerable knowledge on management and doing business, this project will give some challenges as well as providing the organization with much practical project experience, and many will be paying attention.

2.2 Purpose of the Project

NHH's IT Department initiated the Portal Project in order to improve ease of access to electronic services for both students and employees by merging as many of the daily used services as possible. The project will release a tool, which integrates these functions and tools in an operation portal in terms of a web-based desktop. The vision is to enhance and simplify everyone's benefit and appreciation of NHH's knowledge base. Moreover the objective is that the operation portal will be an important tool for interaction and knowledge internally as well as externally.

Today students and employees logging on to a computer at NHH get a standard Microsoft Windows desktop, without any tailor-made tools. However, programs and applications available on the Windows Start Menu are somewhat relevant to the user. This is simply just programs and applications, not services provided by NHH such as it's learning (for course information and interaction), Student Web (for student registration and course selection), or other portals.

In the future, one imagines that once logged on a computer at NHH a desktop will give direct access to relevant tools such as: email, address book, calendar, it's learning, publishing on www or blogs, travel bookings, registration of travel expenses, financial reports, procedurals, and so on. It is clear that all users does not, and should not require access to all these tools, thus only tools relevant for the user will be given access to and displayed in the portal. Information overload will be reduced by more targeted information, such as not giving master students information relevant for bachelor students only (Marthinsen 2008a). Students will receive information based on their courses and field of study. Similarly, employees will receive information based on department and team memberships. So far the project has defined three user groups: researchers, students, and administrative staff, which each will get a tailor-made web portal. In addition each user can customize their portal even further to suit their own personal needs.

2.3 Project Development Method

As suggested by Bouvet in their offer, the Portal Project has adapted an agile Scrum development process in the project. They do, however, not follow Scrum theory 100 %, and prefer to call their method “Scrum Light” due to the fact that several elements in the Scrum methodology have been left out. The project does however keep the most essential elements, such as the use of sprints, daily meetings, and a backlog.

Each sprint starts with a sprint planning meeting and ends with delivery and a sprint review meeting before the next sprint starts. Each sprint lasts four weeks.

The daily meetings are between NHH’s sub-project manager for the portal part and the two programmers. They are conducted in the morning and function as simple progress updates and lasts about 15 minutes, however can occasionally take longer if needed.

There is an overall backlog with items added and prioritized by NHH. This backlog is broken down into an even more detailed and technical backlog for the programmers, so that each task or item takes no more than eight hours, but preferably a lot less.

The project is also divided into phases. The first phase consists of six sprints, and is scheduled to be finished by the summer. Its end is marked by a fully functional operation portal ready to be used by its users, although limited on functionalities, as these will be developed and considerably increased in the next phases.

2.4 Project Organization

NHH’s IT department is responsible for the development and support of the computer and communications systems for research, teaching, and administration at NHH. They provide user support for staff and students as well as having a main responsibility for the information systems, infrastructure, and networks. (NHH Web Site 2009) Bouvet ASA is a Scandinavian consultancy firm specialized in development- and counseling services related to information technology. The company was established in 2002 after some mergers and acquisitions. Today the company has more than 450 employees located in 9 offices in Norway and Sweden. (Bouvet Web Site 2009)

The Project receives its mandate from NHH's management and administrative board, and its sustainability depends solely on their providing funding for the project in the school's budget. The internal project team consists of the project manager, and three sub-project managers for the accompanying projects parts: content, design, and process. The last part has however temporarily been put on hold due to budget limitations. There is also a steering group controlling the project as well as a project support team. The external project team consists of Bouvet's project manager, two programmers, and one graphic and interaction designer. Further KPMG and Inspera AS have also had some involvement in the project.

2.4.1 Project Managers

There are two members of the project who bears the project manager title. Although the IT Department's project managers might be looked upon as the project's true project manager, Bouvet does as well have their own internal project manager. NHH's PM's responsibility is to have an overview and responsibility of everything that goes on on NHH's part of the project, likewise Bouvet's PM has a similar role on their part, in addition to collecting information on progress, and informing and presenting this to NHH. The Bouvet PM is currently located at NHH in an office in the IT Department.

2.4.2 The Content Part

The content part of the project is, as the name indicates, responsible for all content and information to be displayed in the portal, and how. This is understandably not a small task, as there is a lot of information to be handled. The responsibility includes the look of the portal, interaction design, and navigation. It is important to find ways of making the right information easily accessible through high user friendliness and a logical structure. One must keep a close relationship to representatives from the different user groups in order to know what tools and information are of most relevance to them. Since the project's purpose is to improve ease of access to information and integrate resources and tools, the quality of the content part might prove critical to the project's success. While the two other sub-project managers comes from the IT Department, the sub-project manager for the content part comes from NHH's Information Department.

2.4.3 The Portal Part

The portal part is the technical part of the project, as it is responsible for the development and programming of the portal itself. Requests on functionality and design coming from the two other project parts and the project team as a whole is submitted to become the portal part's responsibility when decided upon. The sub-project manager for the portal is the main point of communication between the IT Department and the programmers at Bouvet. The person is also responsible for keeping an overview on the technical requirements as well as to stay up to date on development progress and meeting with Bouvet's programmers on a daily basis.

2.4.4 The Process Part

Although having been put on hold, the process part of the project will most likely come back into the project at a later stage. There are a large number of processes at the institution, many which have been automated in information systems already, such as for instance StudentWeb, which has given considerable work relief to the Student Administration (Danilova and Danilov 2008). Yet there are many more that could either be automated or simplified in an information system, or even increase efficiency by simply integrating those existing into the new portal. It is well known that ERP systems generally require processes to change within the organization in order to give return to the investment, in order to not only be a very expensive auditing system (Dahl and Fuglseth 2007). In the same way, one would expect the portal project to pay off even more if it can improve processes in the organization, although ease of access to integrated information will lead to efficiencies as well. Generating a list of processes was worked upon early on in the project, but has been discontinued. Some processes on the list will be solved nevertheless due to functionality which will be available in the portal anyway.

2.4.5 The Steering Group

The steering group's main purpose is to control the project and making sure the project deliver something that is needed and in accordance with the purpose of the project. Members of the steering group include the project owner and vice-rector, assistant director, professor and dean for the doctoral program, chief of information, and a student representative (Marthinsen 2008b).

2.4.6 The Project Support Team

The Project Support Team consists of three members, all with either considerable IT knowledge or experience from similar project work. One member is the assistant head of IT, the two others are scholarship holders in the Department of Strategy and Management (Marthinsen 2008b). These members serve as available resources which can contribute with their knowledge and expertise when needed, as well as to asking critical questions about the way things are done or when decisions are being made.

2.4.7 Programmers/Developers

In Bouvet there are two programmers or developers working full-time on developing the portal. The chief programmer does most of the actual coding or programming, whereas the other programmer works on designing and planning the architecture of the portal. This includes making an overview of how the different parts of the system and the underlying databases communicates and work together, as well as to be responsible of security and permissions in the portal. In addition to coding and design, the programmers must also write documentation and do testing regularly as elements of the portal are being developed. Both programmers work together next to each other in the same room. There was a third programmer involved 50 % in the process at an earlier stage of the project, but this person has been withdrawn from the project due to budget constraints.

2.4.8 Graphical and Interaction Design

Bouvet also contributed with a graphical designer and an interaction designer. The interaction designer's responsibility is to draw and illustrate where items like logos, buttons, boxes, and text should be located in the portal, and what should happen when clicked on. Based on this the graphical designer will draw and color the actual layout, buttons, and other graphics. This requires only limited time of involvement in the project and they are involved in other projects as well. Due to one designer's resignation from Bouvet, the remaining designer currently does both tasks.

2.4.9 KPMG

KPMG has so far only been involved in the beginning of the project, as they helped analyzing and generating the list of processes which the portal might help with when done. Since the process part is put on hold, KPMG, is no longer involved, but expected to be brought back in to the project when the process part is being resumed.

2.4.10 Inspera

Inspera had initially a role as the second sub-supplier, but is currently no longer involved in the project. They could provide some interesting indexing and processing solutions, which may also prove to be more important at a later stage depending on the project's continuation.

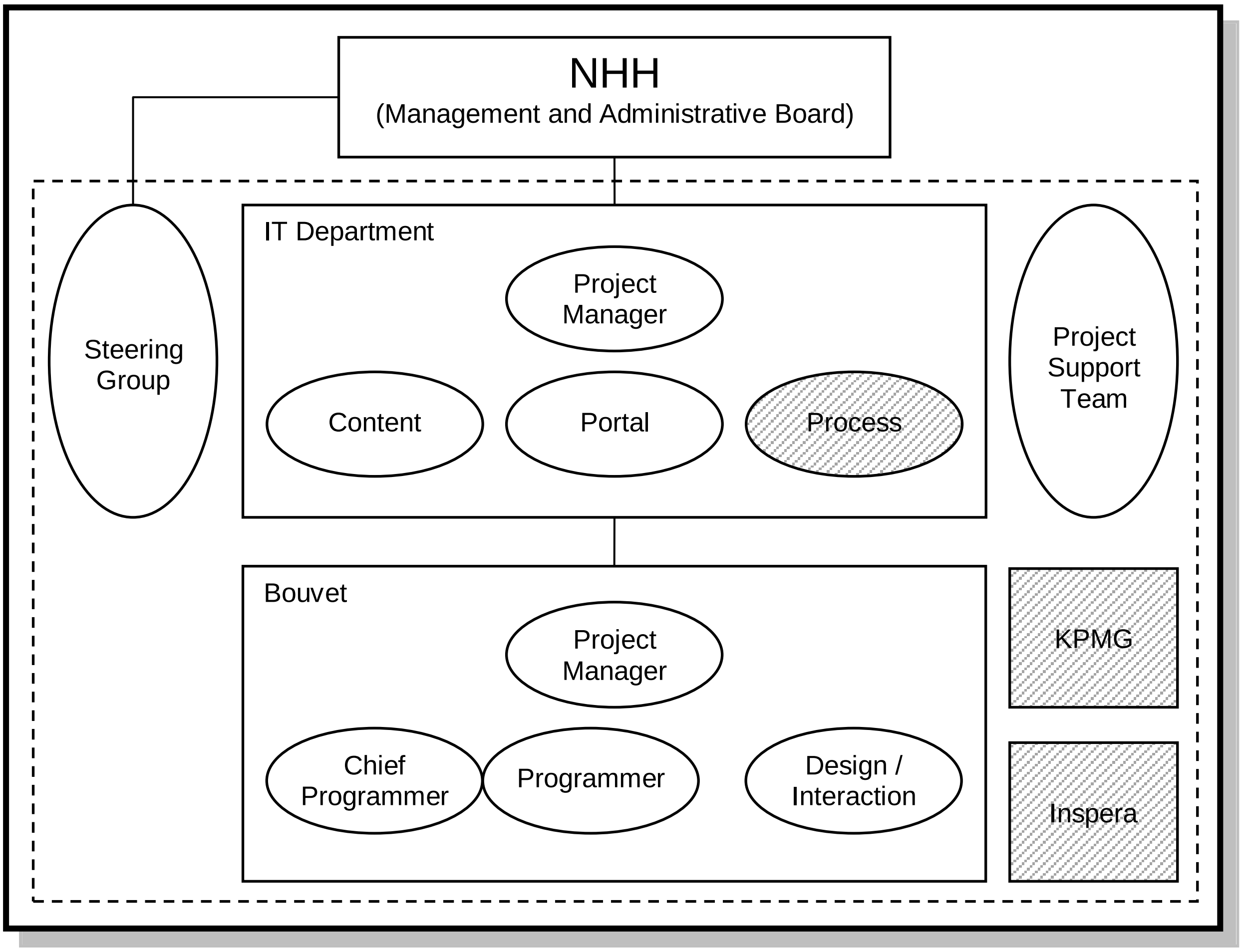

Figure 1: Illustration of the Portal Project Organization (Source: Interviews)

2.5 Current Status of the Project

As stated earlier, this thesis is looking at a project which is still under development. This means that the programmers are still doing coding while this thesis is being written. A number of meetings occur every week. New challenges are met and solved, and bugs are found and fixed. The project is constantly changing and there is no doubt that so are the project members' views on the project. Thus it is important to state that interviews were conducted and most of the information on the project was gathered in March 2009. At this time the project was about to finish its fourth sprint, and the first delivery of a working demo was just about to be available for testing for project members at NHH. Yet there were two more sprints left before the final end of phase one and delivery of a fully working portal (although limited on available tools). Thus this thesis must make the assumption that we are still in March 2009, although in reality much have probably changed already.

The project appears to be on track, in the way that one can use on track in a Scrum project. Bouvet and NHH seem to have gotten used to working with each other and established the necessary trust. There are however budget constraints, and the process part has, as mentioned earlier, been put on hold, or postponed, but with hopes of being taken back into the project at a later stage.

3. Theory and Model Presentation

3.1 Project and Project Management Definitions

3.1.1 Projects

According to Andersen et al. (2004) a project has four characteristic features. It is a *unique task* designed to attain a *specific result*. It requires a *variety of resources* and is *limited in time*.

Tasks that are usually performed repeatedly in a typical organization can be referred to as processes. A project, on the other hand, is a unique task because it is something the organization has not previously worked on. The specific result of a project could for instance be moving the organization to another location. If this were to happen, involving only one department would not be sufficient. It would be necessary to involve people from different parts of the organization with different backgrounds and experience. It might also require external assistance from specialists. This proves the need for a variety of resources. The project ends at a specific date, in this case that date is when the organization is successfully moved. Thus we also understand why the project is limited in time.

The Norwegian word “dugnad” serves as a simple example of a project. A dugnad is a joint contribution performed by a group of volunteers. A typical Norwegian dugnad could be neighbors gathering to renovate the children’s playground in the neighborhood. Many people with a volunteering spirit will show up ready to contribute in one way or another. Whether or not it is planning, painting, fixing, refilling the sandbox, or simply just serve some snacks, they are all happy to help out. The dugnad serves as a project example because there is a clear unique task: to renovate the playground. The specific result is a new playground which is attractive for the children to play in. Several resources are required due to the need of several volunteers with different skills. And there is a short time limit, which for such a dugnad will typically be just a day or two. It should be stated that a great dugnad always ends with a party, serving to mark the dugnad’s end as well as to rewarding the volunteers. Showing such traditional Norwegian spirit, Holmesland (2007) suggests that all projects should have a party in the end, rather than the project organization simply just dissolving when a project has come to the point of termination.

3.1.2 Project Management

Project management includes four different main tasks: (1) establishing a foundation for the project, (2) planning the project, (3) organizing the project, and (4) controlling the project. (Andersen et al. 2004)

While launching a new project it is necessary to have support from the rest of the organization, so that the people involved are encouraged to make an effort for the success of the project. This includes explaining why the project is necessary, and what the final result is intended to be. If successful in doing this, one has managed to establish a good foundation for future work on the project. Planning includes making a list of what needs to be done and when. The use of several milestones will help ensure that the project is worked upon continuously. The planning needs to be detailed on some levels, but not necessarily on all.

It is very important to make sure that the resources are available when needed; this is simply an organizing task. In addition one must prepare for cooperation between all project participants. The tasks that are to be done must also be delegated. Finally the controlling is necessary not only to make sure that everyone does its job, but most of all in order to notice if the project is lacking behind schedule as soon as possible so that necessary decisions can be made.

3.2 Waterfall Software Development

The concept of Waterfall software development origins from Winston W. Royce’s (1970) article “Managing the Development of Large Software Systems.” However he did not invent the concept, neither did he approve of it, but rather described it as an (inefficient) way large software systems were developed at the time, which needed improvements. Its name indicates a view of a development process in which one flows from one development steps down to another; and there is no way (or at least high difficulties and costs) of moving back up the waterfall. The water can only flow down, not up. This has now become known as a traditional way of software development, or a heavyweight method.

Although different waterfall models differ somewhat, the original seven steps described by Royce are the following: (1) system requirements, (2) software requirements, (3) analysis,

(4) program design, (5) coding, (6) testing, and (7) operations. The waterfall-like process is illustrated in figure 2:

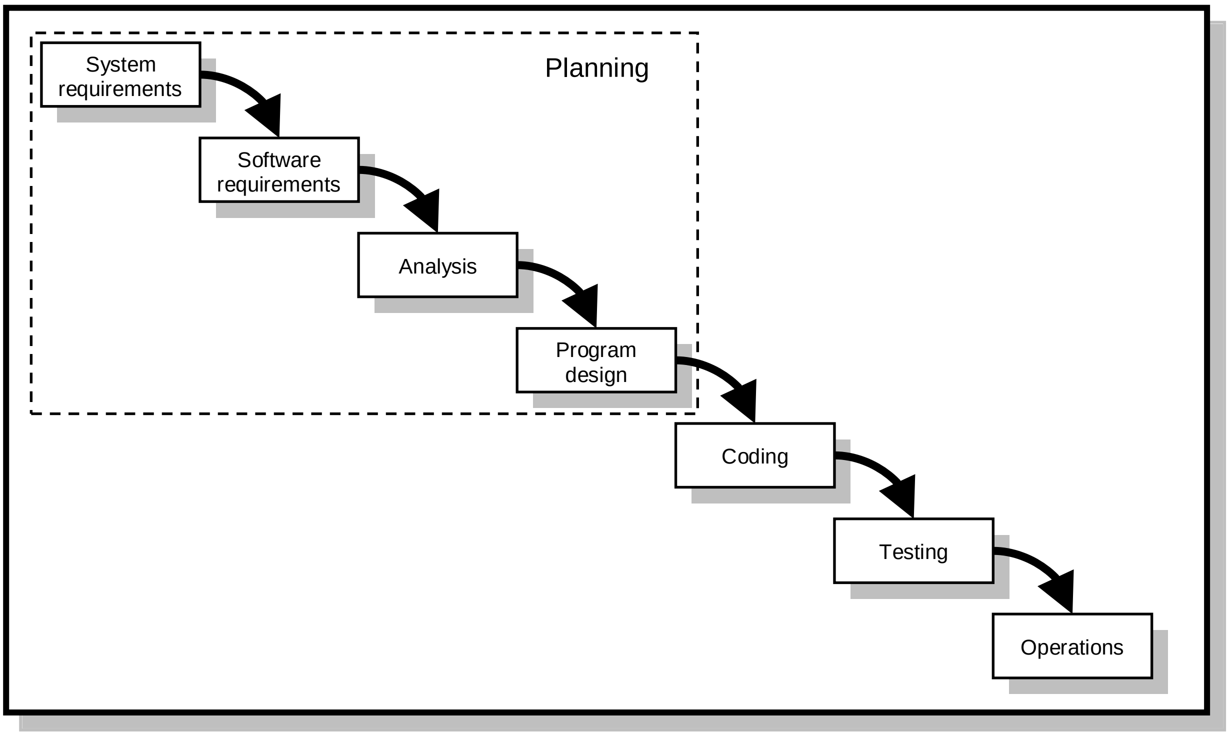

Figure 2: Waterfall Development Process (Royce 1970)

The first state, system requirements involves listing the functionality of the system as a whole, while the software requirements specify how the software itself behaves. Thereafter an analysis based on the requirements is conducted, before the whole system is specified and designed. Once the whole systems is planned and one “knows” what one wants it is time to start programming and writing the code according to specifications and design. When done, only testing is left so that bugs can be fixed, before the system is delivered and goes into operations where only maintenance is required.

At first glance, and with little hands-on IT project experience, this does indeed look like a very reasonable way of planning and working on developing a new IT system. To be honest it looks logical and appears to make sense. On second thought, however, we notice that the first four stages is actually only related to planning, i.e. more than half of the stages in the process. It might be well-know that planning is important and can prove critical in project management literature, but IT projects are different. If you’re building a ship or designing a new car, you want to be sure the ship can float and that the car can drive, and thus focus on proper planning in advance. It is just as important that an IT system works when delivered, thus the stages mentioned above are needed here as well. What waterfall development processes do not take into consideration is the constant change in both technology and the customers' needs. The plans of an IT system may be relatively unclear, so that the final product delivered to a customer is not at all what was expected, or needed. This problem is the reason why one has sought to develop other software development methods which takes this constant change and need for flexibility into account.

3.3 Agile Software Development

3.3.1 The Background for Agile Software Development

Eight years ago 17 people met for a friendly weekend at a ski resort in the Wasatch mountains of Utah in the United States. These were representatives for a range of different software development methods, including eXtreme Programming, Scrum, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming, etc. Despite the fact they were supporters of such a list of different approached, they all agreed about a need for an alternative to the traditional "documentation driven, heavyweight software development processes" (Agile Alliance 2001).

The Agile Manifesto is replacing the heavyweight waterfall model, however, the iterative, evolutionary, and incremental ideas of the Agile Software Development is not as modern as one might think (Larman and Basili 2003). Truth to be told, we are actually going back in time, since this way of development dates back to the mid-1950's, or even earlier. However methods comparable to a waterfall model evolved and increased in popularity. Although Winston W. Royce (1970) (who many waterfall model users refer to) show hints of iterative development, feedback, and adaption in his article, these ideas seems to have been lost in the later evolvement of the model (Larman and Basili 2003). Tomb Gilb (1976) was among the first to promote evolutionary software management already in the 1970s, explaining clearly that: "Evolution is a technique for producing the appearance of stability. A complex system will be most successful if it is implemented in small steps and if each step has a clear measure of successful achievement as well as a "retreat" possibility to a previous successful step upon failure. You have the opportunity of receiving some feedback from the real world before throwing in all resources intended for a system, and you can correct possible design errors...” (Gilb 1976, Larman and Basili 2003:50).

The reason though why the Agile Manifesto evolved is that there is an urge to get back to the old lightweight ways, and get away from the heavyweight waterfall models. Larman and Basili (2003) realize that waterfall models are popular because they are simple to explain and easy to recall, appears to be orderly, accountable, and measurable processes as well to having been promoted heavily in the past. Whatsoever their conclusion and strong recommendation is clear: “In the interest of promoting greater project success and saving taxpayer or investor dollars, let’s continue efforts to educate and promote the use of IID [Iterative and Incremental Development] methods.” (Larman and Basili 2003:55)

3.3.2 What is Agile Software Development?

Agile Software Development consists of a group of project management techniques taking into consideration how software development and IT projects require another management approach than most other projects. In short, agile theories focus on getting started with producing code early on, rather than spending too much time on planning and documentation. This enables the project to be more adaptable to change. Working code is what the customer needs, and thus this is what one measures progress on, and want to deliver as soon as possible. As the first principle states: “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.” (Agile Alliance 2001).

3.3.3 An Overview of Some Agile Methods

Derived from Wikipedia (2009a) we come to the following short descriptions of some of the agile development processes:

eXtreme Programming is a method prescribing a set of daily stakeholder practices and encourage the use of a list of values. It is believed that following these practices will lead to a development process that is more responsive to customer needs while creating software of high quality.

Scrum is a process which use a set of practices and predefined roles. The primary role is the Scrum Master which is similar to a project manager. Development is done in iterations called sprints, lasting 2-4 weeks with short daily meetings, working on tasks prioritized in a backlog list.

DSDM is an iterative and incremental approach that emphasizes continuous user involvement. Its goal is to deliver software systems on time and on budget while adjusting for changing requirements along the development process. There are 9 underlying principles consisting of four foundations and five starting-points.

Adaptive Software Development replaces the traditional waterfall cycle with a repeating series of speculate (planning), collaborate, and learn cycles. This dynamic cycle provides for continuous learning and adaptation to the emergent state of the project. The characteristics of an adaptive software development life cycle are that it is mission focused, feature based, iterative, time boxed, risk driven, and change tolerant.

Crystal focuses on efficiency and habitability as components of project safety, as well as focusing on people, instead of processes or artifacts. The method requires the following properties: frequent delivery of usable code to users, reflective improvement, and osmotic communication preferably by being co-located.

Feature-Driven Development blends a number of industry-recognized best practices into a cohesive whole. These practices are all driven from a client-valued functionality (feature) perspective. Its main purpose is to deliver tangible, working software repeatedly in a timely manner.

Pragmatic programming defines a pragmatic programmer as one who is an early adopter and fast adapter, inquisitive, critical thinker, realistic, and jack of all trades. As a pragmatic programmer you do not stay loyal to only one of the above methodologies, but choose the methods and techniques that work best for you (Dinkla 2009).

3.3.4 The Agile Manifesto

The Agile Manifesto (Agile Alliance 2001) states four values for agile development:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

The manifesto does not ignore the items on the right, but makes it clear that the items on the left should be prioritized if put in a situation where we must choose one over the other.

Frequent interaction between individuals compensates for minimizing documentation (Highsmith and Cockburn 2001a). It promotes sharing of information and allows for changing the process fast when needed.

By measuring progress on working software we can easily see how fast we can produce results. Further it allows us to see how the system is currently working and provide quick feedback.

Customer collaboration over contract negotiations shows us the importance of cooperation with all parties involved. In order to develop a successful project everyone must work together toward the common goal. This allows for misunderstanding and problems to be solved early. Contracts are understandably necessary, but it is collaboration which will lead to success (Highsmith and Cockburn 2001a).

Many will say planning is essential in order to manage a project well. However if frequent changes in the project environment occurs, it is more important to respond to the changes, rather than following the initial plan. If new information tells us that we should do something else than what we are currently doing, then we should change what we are doing, even though the plan tells us otherwise. You will never hear a battle commander say "we lost the battle, but we were successful because we followed the plan to the letter" (Highsmith 2002).

This does not mean that we should not plan at all. We can make simple plans, but if changes happen we must acknowledge that the plans might no longer be valid and needs to be changed. One may also suggest to plan for change, i.e. knowing where uncertainties exist and have different plans for different scenarios the may occur. I will not say that this cannot prove to be useful, but I do believe it is hard to predict what sudden technological changes may appear during system development. One can to some extent forecast how processing power and memory size will increase, but ability to predict in advance how our system should adapt to a currently unknown technological innovation which suddenly appear out of nowhere is rather impossible and a waste of time. How could for instance engineers working on improving technology on developing old-fashion 35 mm films ever plan their projects to be prepared for a sudden launch of digital cameras when such a thing was unheard of? The answer is simple: they could not. Whatsoever, they should not continue their projects as though nothing happened.

I guess no one would ever start on a project without any plans at all, so we need some sort of planning. In agile development there will be overall planning in the beginning, and then short time-framed plans that can easily be changed during development.

3.3.5 Principles behind the Agile Manifesto

The people behind the Agile Manifesto have in addition to the manifesto itself, also listed twelve principles to be followed in order for a project to comply with Agile Software Development. The principles are as follow:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity—the art of maximizing the amount of work not done—is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Source: http://www.agilemanifesto.org/principles.html

These principles give a great impression of what Agile Software Development is all about. They can serve not only as principles, but also as guidelines for how to manage an IT project in an agile way in practice. The first principle emphasize customer satisfaction as the ultimate goal, and the rest shows us how and what to keep in mind.

3.3.6 Criticism of Agile Software Development

I find little criticism of Agile Software Development in articles, and the people behind it are a group of outstanding peers with considerable experience on the field. They come from supporting different development methods, yet they have all agreed upon all these values and principles with few disputes. I do whatsoever claim that the method appears too technically focused, perhaps due to being developed mainly from American management culture? Holmesland (2007) claims PSO is project management, the Norwegian way. Although this does not mean they say the non-technical perspectives are not important, there is a lack of attention to these, which I see as just as important aspects in IT project management. Therefore we should also recognize the need of a PSO perspective, which in addition to a technical (system) perspective also try to focus equally on the people and the organization as well. Further it should also be mentioned that that there is a lack of empirical research comparing agile software development with other methods. Turk et al. (2002) points out that empirical data comparing agile and non-agile approaches would greatly enhance our understanding of the true benefits and limitations of agile processes.

3.4 Scrum

Scrum is a popular iterative and agile development process which is often being credited to Ken Schwaber (1995) due to his article “SCRUM Development Process.” However this kind of process was not entirely new, and had been used earlier (Sutherland 2004). The name is not an acronym, although often spelled in capital letters, but is simply just name derived from rugby football. It is “a tight formation of forwards who bind together in specific positions when a scrumdown is called” (Schwaber 1995). The method can be characterized by its sprint iterations ensuring continuous coding and deliveries, the use of a backlog for priorities, and defined roles for those involved. Scrum’s goal is to deliver as much quality software as possible within these short sprints. (Beedle et al. 1999)

3.4.1 Scrum Methodology

The Scrum methodology consists of three main groups of phases: pregame, game, and postgame (Schwaber 1995). Pregame is the first phase of a development project and includes planning, system architecture and high level design. The postgame is the closure of the project and comes in the end. Both the pregame and postgame has similarities to a traditional waterfall model’s list of steps to be followed in a fairly linear flow. The game part is the major phase in which the project will be in for most of its life cycle, and is what makes Scrum an iterative process and successful way of system development. The game consists of a set of sprints, i.e. iterative development cycles, which ensures constant evolvement of the project. Figure 3 serves as good illustration of how the iterative game occurs after and before the two other linear phases. There are also three defined roles: the scrum master, the product owner, and the development team (Schwaber 2007).

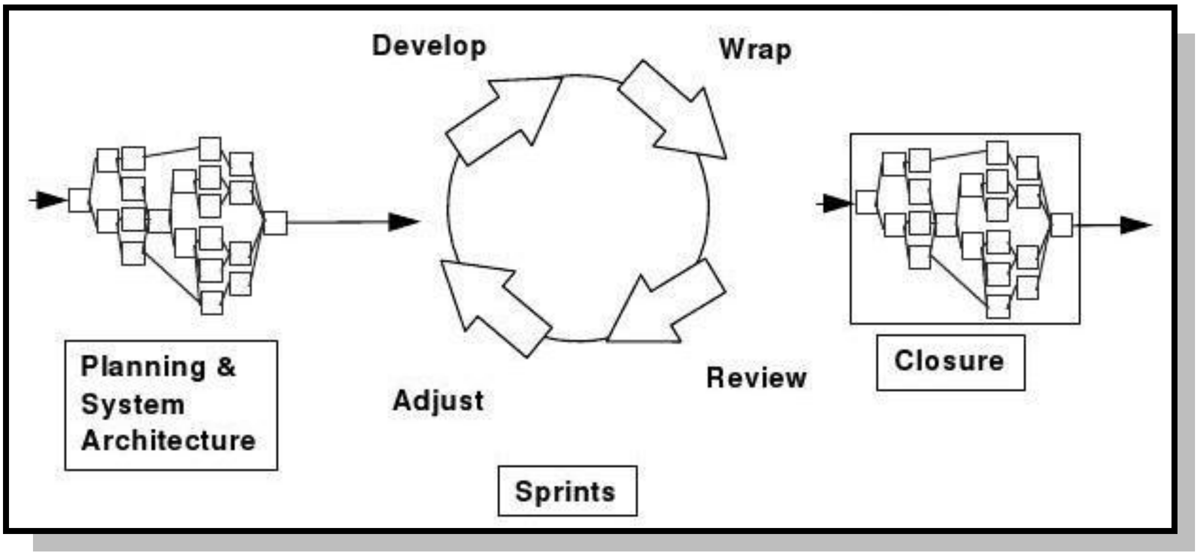

Figure 3: Scrum Methodology (Schwaber 1995)

3.4.2 Scrum Roles

The Scrum Master, the Product Owner, and the development team are the three roles which hold responsibilities for project management in a Scrum project. The Product Owner represents the interest of all stakeholders in the project and is also responsible for the success of the project. He or she is also responsible for initial and ongoing funding by providing the project requirements through the Product Backlog and ensure that the most valuable functionality is worked upon by frequently prioritizing the backlog. The development team is self-managing, self-organizing, and cross-functional. They are responsible for developing functionality incrementally throughout each sprint. The success of each iteration is the team's responsibility. The Scrum Master is responsible for making sure the project is a Scrum project, and that the Scrum methods are adhered to. This responsibility is taken by teaching project members about the Scrum method, making sure the method is implemented in a way that fits the organization and its culture, and ensuring everyone follows Scrum's rules and practices (Schwaber 2007).

3.4.3 The Pregame and Planning

The pregame phase of planning and system architecture includes a list of steps that need to be gone through before the project is ready to enter the game phase. Schwaber (1995) provides a comprehensive list of these steps. Generally the process looks very much like the prephase of any project, and includes making a list of what one want the project to deliver (i.e. backlog), defining teams, assess risk, estimate costs, and receive management approval and funding. Thereafter the architecture/high level design process is more technical and focused on reviewing and changing backlog items, as well as to identify problems or issues which may occur in development.

3.4.4 The Game and the Sprints

Tom Gilb's (1988) book “Principles of Software Engineering Management” was the first book giving substantial dedication to discussion and promotion of iterative and incremental development (IID) and awareness of IID accelerated significantly in the 1990s (Larman and Basili 2003). There is no doubt that Scrum has been affected by IID, and thus is a form of IID when considering the Scrum sprints. The sprints constitutes the game phase of scrum and is the essential part that distinguishes Scrum from other development processes, as it is in the game, and by the use of sprints, development occurs, code is written, and releases launched. The sprints occur iteratively every one to four weeks until the product is deemed ready for distribution (Schwaber 1995). However, about a decade later Schwaber (2007) is leaning more toward that sprints should be lasting one month instead.

Schwaber (2007) further explains that between each sprint there should be a sprint planning meeting and a sprint review meeting. Each sprint is initiated with a sprint planning meeting where the product owner and the development team discuss and agree about what should be done in the following sprint. The sprint planning last one day and is divided in two. The first part includes the product owner presenting a new and updated product backlog of the highest priority items for the next sprint. The development team then asks questions about each item until they understand and know enough about it. This illustrates the importance of customer collaboration as described in the Agile Manifesto. In the second part the development team selects as many product backlog items they believe can be developed in the following sprint. A tentative plan of how work can be done is made, as well as to generate a more detailed sprint backlog. Figure 4 illustrates how features in the product backlog are put into the sprint backlog, before the sprint starts, and ultimately ends with delivering more working code at the end of each sprint:

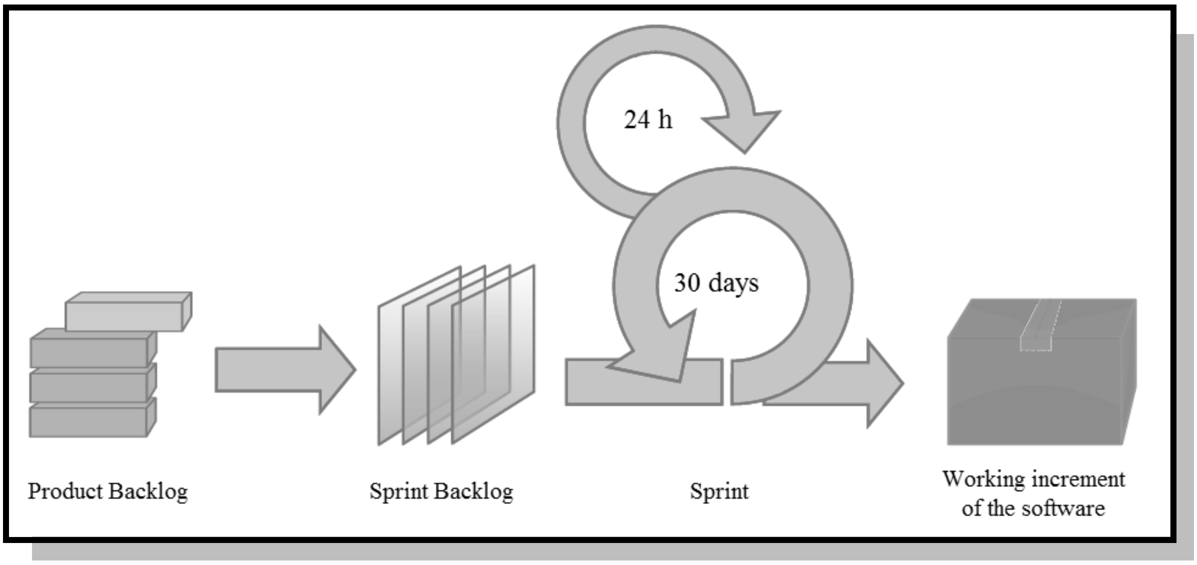

Figure 4: Scrum Sprint Model (WikiPedia 2009c)

The day after the Sprint Planning, the new Sprint has started and developers are back in front of their screens writing code, and ensuring progress. Early every day the development team meets for only 15 minutes. This is called a daily scrum and is mainly about the development team answering the following three questions (Schwaber 2007):

- What have you done on this project since the last daily scrum meeting?
- What do you plan on doing on this project between now and the next daily scrum meeting?

- What impediments are in the way of you meeting your commitments toward this sprint and this project?

The purpose of the daily scrum is to “synchronize the work of all team members daily and to schedule any meetings that the Team needs to forward its progress. The team members are inspecting each other’s work in light of the team’s commitments, and making adaptations to optimize their chance of meeting those commitments.” (Schwaber 2007:3)

At the end of each sprint one day is dedicated to a four-hour sprint review meeting and a three-hour sprint retrospective meeting. During the first meeting the development team presents to the product owner, and any other stakeholders interested in attending the meeting, what was developed during the sprint. This is an informal meeting where product owner and stakeholders may inspect the development team’s work, and all may collaborate about what to do in the next sprint based on what was just completed. Thereafter the scrum master holds a sprint retrospective meeting with the development team. In this meeting the development team is encouraged to revise their work and practices within the Scrum framework in order for the process more effective and enjoyable in the next sprint.

Altogether the sprint planning, the daily scrum, the sprint review, and the sprint retrospective meetings are the elements which constitute the inspection and adaption practices within Scrum (Schwaber 2007).

3.4.5 The Postgame and Closure

When the management team feels that the product, or a new version of it, is ready to be released, a new sprint will not be initiated; instead one enters the postgame phase, i.e. the closure of the project. One prepares the product for a general release. Final testing, user documentation, and preparation for trainings, are examples of tasks which will be performed during closure. When ready, the new product will be released.

3.4.6 The Backlogs

I have briefly indicated that Scrum uses backlogs as lists of requirements for the project. There are two kinds of backlogs; the product backlog and the sprint backlog.

The product backlog functions as a wish list of functionality from the customer. This list is never complete; it is dynamic and subject to change at any time during the project, and the product owner is responsible for updating it and adjusting the priorities. Ability to make changes under way ensures the agility of the Scrum method. A product backlog includes an item name, initial estimate, complexity factor, adjusted estimate, as well as information on which it will be developed. The complexity factor increases the adjusted estimate due to project characteristics that reduce team productivity.

The sprint backlog on the other hand is the responsibility of the development team, and one creates a new backlog at the beginning of each sprint. The sprint backlog is based on the items from the product backlog to be developed in the sprint. Each item in the product backlog is represented as several items in the sprint backlog. The development team creates the sprint backlog by making a list of technical work and tasks needed to be done in order to deliver the items on the product backlog. Each task should take about four to sixteen hours to finish. If a task takes longer than this, it must be broken down to smaller in order to fit within this time requirement. The sprint backlog serves as a real-time picture of what and how much has been completed during the sprint. Each task shows who is responsible, whether it is completed, in progress, or not started, as well as a time estimate and how much time is left.

3.4.7 The Burndown Chart

Information on items in both the product and sprint backlog will show how much time remains until completion for each item. Based on this information one can be easily sum up amount of work remaining at the end of each sprint, or day. This can give us a clear visual graph showing the progress of the project (Shojaee 2008). This is called the Burndown Chart. It can also allow us, based on the slope of the graph, to estimate in which sprint all work will be done. See figure 5 for an example of a burndown chart:

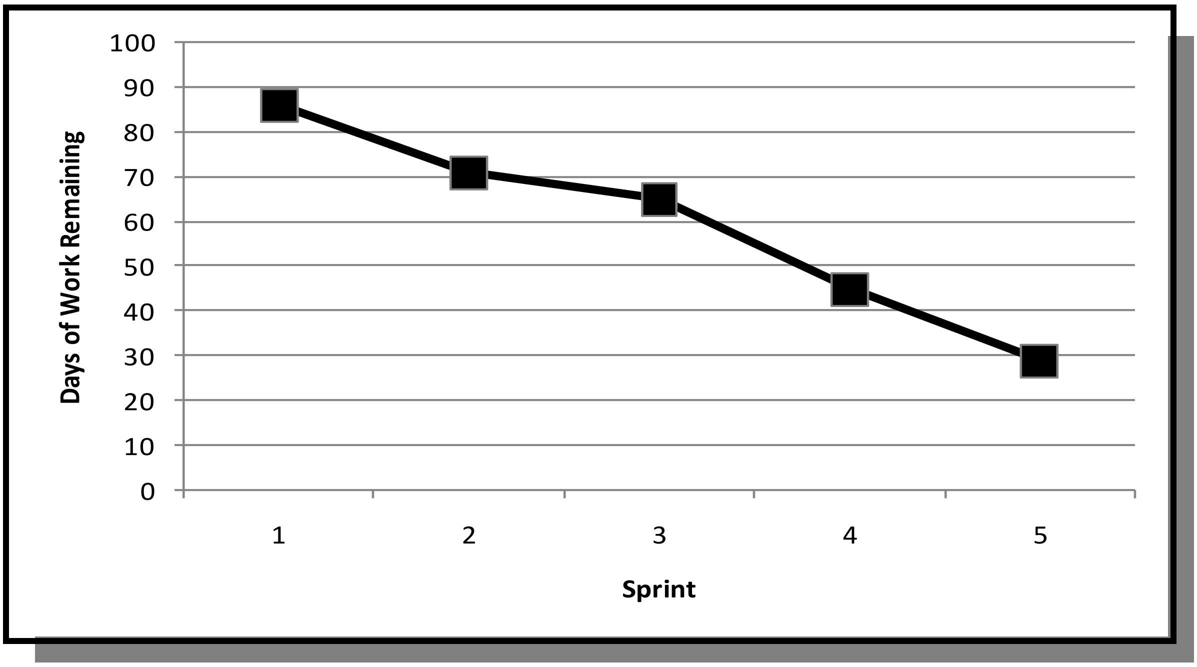

Figure 5: Sample Scrum Burndown Chart (Schwaber 2007)

3.5 The PSO Perspective

Although the agile development theories claim to take customers into consideration, and focus strongly on customer collaboration, this is true mainly on the system- or software perspective. Andersen et al. (2004) give in their Goal Directed Project Management (GDPM) a strong focus on planning and making milestones – which contradicts somewhat with agile methods. I do whatsoever believe that their PSO perspective is indeed very important especially in IT project management as well, and tend to be left out in the agile theories.

3.5.1 The Background for a PSO Perspective

PSO projects – abbreviated from development of people, system, and organization – is a result of a lesson learned from considerable experience from many IT system implementation projects (Winter et al. 2006). This is a way of thinking that needed to be developed due to the nature of technical project participants' nature of becoming too technically oriented. Grude (2007) explains a major challenge in IT projects being that software engineers like to talk to computers, but working and managing an IT project requires you to know how to talk to people, which tend to respond quite differently than computers.

Being too technically oriented, means focusing too much on the technical part of the project. Related to IT projects, it means that most of the focus in the project concerns the IT systems itself, its design and functionality, and the development of it. According to Andersen et al. (2004) it is easy to be too occupied with the system so that training and motivation of people is forgotten, as well as to completely overlook the fact that the system might facilitate completely new forms of organization. One might be very much concerned about making sure that the system can fulfill all the requirements, making sure that it uses the latest technology, and adding advanced features. Developers will make sure that the system deliver all specified technological needs and try to deliver a "perfect" solution. Yet, the problem is that even though a "perfect" system is delivered, the project, and thus also the system, can prove to be a complete failure.

If the organization is not prepared to adapt to the new system (if necessary), or if the organization does not know how to change their current processes, or simply just cannot change existing processes, it does not matter how well the system is designed, it simply is not possible to use. If the people in the organization have not been trained in using the system, or if the system has not taken the people's preferences into account, the system will not be taken advantage of properly. It does not matter if the system can do a variety of new things, and automates processes, if the most used features are hard to find, the not-so-important features are easily accessible, and the advanced properties are hard to understand, we will not get the efficiency expected, it might even prove otherwise. Further, if the staff is satisfied with current system and show resistance to change we have yet another problem.

3.5.2 What is a PSO Perspective?

A PSO perspective serves as a reminder for us to not only think about the system part of the project. We need to think of the project as three-folded with three different elements, the system being only one of them. The two other elements, people and organization, are just as important and require equal amount of attention. The idea is to keep a balance between all elements (Andersen et al. 2004). I prefer to call this lack of focus on the people and the organization a pSo perspective, as opposed to a PSO. Compare illustrations in figure 6 and 7.

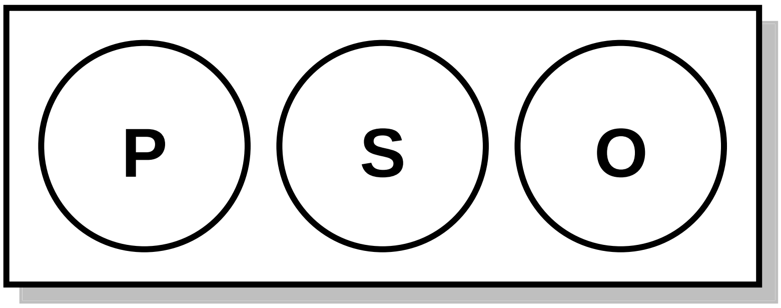

Figure 6: PSO Perspective (Andersen et al. 2004)

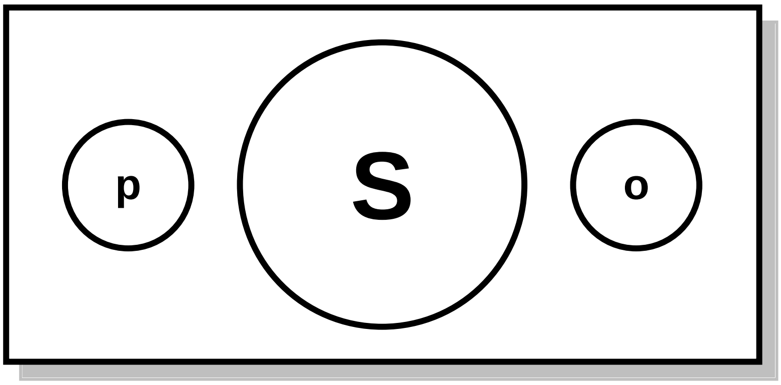

Figure 7: pSo Perspective (not PSO)

The PSO perspective should be taken into consideration in as many aspects of the project as possible. Andersen et al. (2004) shows us that linking the milestones to each of the PSO elements can lead to great success. One can for instance have one path of milestones for each element. Thus we will not only have milestones related to the system, but also sets of milestones related to both the people- and organization perspectives as well. Just like we would normally have milestones depending on each other, or having to be met in a specific order, we can with PSO milestones make sure that tasks related to people and organization are performed before one goes to the next level of system development. If a GANTT chart shows us that a survey among employees, and middle-managers' feedback must have been received before we can start writing a specific part of the code, we know the people will be taken into consideration.

The PSO perspective is closely related to Leavitt's Diamond (Leavitt 1965, Andersen et al. 2006). Leavitt's Diamond is defined by BNET (2009) as "a model for analyzing management change [...] Leavitt's Diamond is based on the idea that it is rare for any change to occur in isolation. Leavitt sees technology, tasks, people, and the organizational structure in which they function as four interdependent variables, visualized as the four points of a diamond. Change at any one point of the diamond will impact some or all of the others. Thus, a changed task will necessarily affect the people involved in it, the structure in which they work, and the technology that they use. Failure to manage these interdependencies at critical times of change can create problems."

3.6 Agile Software Development vs. PSO

As mentioned earlier the agile methods lacks on the people and organization elements, and the GDPM's focus on planning and milestones contradicts with the need for agility in IT projects. To defend the agile theories, these are mainly considering the software development, and not the project management as a whole, and to defend Andersen et al (2004), GDPM does suggest not planning further ahead than what is practical and needed, giving some room for change and thus agility. The PSO perspective has, as stated earlier, been developed as a result of many too-technical-focused IT projects. Now, the Agile Manifesto and principles do mention the great importance of people, but my view is that the agile methods focus on people working on the project, and coordination amongst them, whereas people in the PSO view means take into consideration all the people that will be affected by the project, but who do not necessarily have influence on the project itself.

The agile development might be a way of succeeding in the system perspective, but we must not forget about the people and the organization. All three should be given equal importance from the project management point of view. The project will not be a success if only the system is perfect, if the people cannot use it, or the organization is not adjusted to fit it. The PSO concept "forces one to think through consequences and possibilities in fields other than the purely technical." (Andersen et al. 2004:4) I am convinced that adding this perspective to agile development will prove very valuable.

3.7 Why are IT Projects Late and Delayed?

IT projects usually end up as either complete failures or huge successes. The failures might be delayed and delayed for a long time, ending up either delivering a non-working product, an out-dated product, or something no one can or want to use (lack of PSO), or simply nothing at all. Huge successes has delivered exactly what one wanted, and maybe even more, as well as managing to contribute to the organization so that it can take advantage of more efficient processes leading to possible cost-efficiencies and competitive advantages.

According to Brooks (1995) IT projects are late for the following five reasons:

1. Techniques of estimating are poorly developed.

2. Effort is confused with progress; men and months are not interchangeable.

3. Urgency affects estimates of how much time tasks requires.

4. Schedule progress is poorly monitored.

5. When schedule slippage is recognized, manpower is added.

Brook's main point, since already in 1971, is that thinking about man-months in IT projects is unrealistic because resource requirements for many tasks in an IT project just cannot be measured by man-months. There are things that simply take a certain amount of time regardless of the number of people working on it. Thus if we have delays, adding manpower is not the solution. We must either realize that the project will take longer, or cut down on requirements. Adding manpower can actually result in the project taking even longer, for instance due to training and intercommunication.

Programmers also tend to be very optimistic, and early estimates are highly optimistic, rather than realistic. They are best-case-scenarios, not taking into consideration testing and bug fixing. Brooks has been successful with the following rule of thumb for software scheduling: 1/3 planning, 1/6 coding, 1/4 component test and early system test, and 1/4 system test, all components in hand. Andersen et al. (2006) emphasize that over-optimism with projects is dangerous and suggest that those who will do the work must participate in the planning; and those who are responsible for implementation must be realistic.

As a result of his own experiences and research Brooks has come to introduce the following law in IT project management:

> *"Adding manpower to a late software project makes it later."*

Scrum development works around this with its very strict view on time. There must be a scrum meeting every day with a progress update, thus some development must be done every day, and there must be a delivery by the end of each sprint, lasting a specific number or days, and no more. This allows an agile scrum project to having delivered at least something, and at least the most important functionalities, before project members can even start feeling there might be a delay in the project. A scrum project will deliver a product by the end of the project, no matter what. The requirements and requested functionality in the product backlog keeps changing all the time, and the list will most likely even grow, making it hard to develop all the functionality which the customer may come up with during the project. Due to the nature of the re-prioritization of the product backlog before each sprint, the scrum method ensures having delivered the items with highest priority, and as much as possible when the project is set to deliver its product. If there are still items remaining in the backlogs, but many items having been added along the way, I will leave it up to others to define whether or not this is a delayed project if one wants to keep it lasting for a couple more months in order to add more functionality. Highsmith (2002) is convinced that agile projects will deliver considerably faster than traditional projects. And if not all can be delivered when expected, due to the prioritization focus on producing code a product serving the most important need will be available early anyway.

4. Methods and Techniques

The purpose of this chapter is to give a brief overview of how this study has been conducted, and reasons for selecting the research method which has been used.

4.1 Qualitative Method and Case Study

This master thesis is a qualitative research paper, and is based on qualitative data instead of quantitative. As opposed to quantitative research which is “the gathering and analyzing of measurable data,” (CPA 2009) quantitative research can be described as “the gathering and analyzing of data based on interviewees’ own perceptions or experiences in order to provide insight into their beliefs about their circumstances rather than measurable data.” (CPA 2009)

I am doing my research on a small IT project which applies some of the theories I wish to learn more about, and thus it is a research where I acquire my knowledge from a specific environment, a situation which according to Repstad (1999) is well fitted for a qualitative research method.

My research has been conducted as a case-study, which has allowed me to look deep into a real-life example of an IT project. I have entered the project organization and tried to acquire as much information as possible within the time available. This has given me the opportunity to get insights from the project that could not have been acquired in any other way. Having practically no relationship or connection to the project I am in a position to make a more objective analysis of my findings. I am indeed a student at NHH, but since the portal is not to be launched until after my graduation, it will not affect me, and should not have any greater influence on my doing an objective analysis. It could however somehow affect my research in the way of looking at the portal too much from a student’s point of view, but I have tried to be aware of this challenge, and assume that it has not been of significant influence on my thesis.

4.2 Interviews

I have performed my case-study by doing semi-structured interviews with people involved in the portal project. Interviews are the most commonly used method in qualitative research, and semi-structured interviews are flexible in the way that they only follow themes to be explored during the interview instead of following a list of question, and they allow new questions to be brought up during the interview based of what is being said (WikiPedia 2009b). Thus it feels more like an informal conversations which is about a specific theme, where a list of interesting topics have been prepared in advance by the interviewer to make sure no topics are forgotten in the conversation.

I have conducted five interviews with a total of six members of the portal project organization; four members which are employed by NHH, and two which are employed by Bouvet ASA.

Each interview was conducted using a tape recorder, instead of taking notes, allowing the situation to be less formal. No interview object showed any resistance to being recorded, and I got little impression of the tape recorder influencing the conversations. Some respondents would not only sit and talk about the project and answer questions, but also draw to examples, and show documents and how they worked together on the computer screens, as well as to show me a demo of the system.

The interviews were listened through and noted down in keywords and quotations. The most relevant and interesting parts were analyzed further. All interviews were conducted in Norwegian, and the information gathered and quotes have been translated into English to my best knowledge. Anonymity has been offered by using no names in the thesis, only respondents' roles in the project, and the tape records will be deleted once the thesis has been approved.

The main purpose of the interviews has been to learn about the project, get inside information from a practical project, not just theory about it, and of course learning about positive and negative sides with this particular project.

4.3 Validity and Reliability

Validity and reliability is of considerable interest in social science research, and thus I find it necessary to briefly discuss the validity and reliability of this master thesis.

Validity can be separated into internal validity and external validity. Internal validity is according to Merriam (1988) about to what degree ones findings represents the reality; are they real? External validity, on the other hand, is about to what degree one can generalize the findings. A high degree of reliability assumes that the results are so certain, that one would get the same findings even if the research were repeated several times.

Most of the information I have gathered depends solely on the interviews that have been conducted. I trust that the information given by the respondents is the reality, and represents their true feelings and understandings of the topics being discussed in our conversations. The project members' experience with the project is not available from any other sources than themselves, and thus an interview is the best way of acquiring this information. To avoid misunderstandings or misinterpretations I have tried to repeat, in my own words, what respondents have explained in order to receive feedback on my understanding being correct. Although a combination of several methods for acquiring this information would be preferable to achieve higher internal validity, I believe that conducting interviews have been sufficient as they give me information directly from the sources. There is also a considerable share of the project members which have been interviewed, and thus I will argue that the internal validity is satisfactory.

Since this study has been performed on a rather small project compared to many other larger scaled IT projects by international IT consultancy companies or in multinational companies around the world, it would be difficult to say much about how my findings can be generalized to apply for all these other projects and organizations. NHH is a public institution and a school, and not a private profit-making manufacturer or service provider. This will affect greatly the way a project is worked upon, and put limits to what degree the results can be generalized, and we can also see the strong focus on the budget constraints. Thus the external validity may be somewhat limited, but although there is little evidence that results can be generalized, it does not mean they are not. Repstad (1999:11) states that qualitative methods looks deep into what is being researched on, but not so much is breadth. He further adds that one cannot generalize uncritical from a qualitative study

(Repstad 1999:14). I believe however that it can raise many questions for further research. There are many challenges and benefits NHH experiences in this project that are most likely to be present in many other IT projects trying to apply a new and agile approach to development. This, however, might be better discovered in a quantitative study.

I am grateful for all information which I was given, and for those who were willing to contribute to my thesis by making themselves available for interviews. I do however realize that some information must have been left out. This may have been negative aspects which one might not want to be described to the public. This master thesis will be published at the same institution in which the project belongs, and thus will be more easily accessible to anyone interested in the project. This might lead to respondents hesitating on what information to be given as possible increases of the budget relies on the project having a bright future. I have come to realize that the fact that the project is far from done, might be a threat to the reliability of the study. The study should be highly reliable in the way the interviews were conducted by tape recordings and ensuring that answers were understood by me correctly, and thus I believe that having done the same study, at the same point in time of the project, would have given the same findings. Yet one might expect a higher degree of self-skepticism and willingness to reveal more negative aspects after the project's termination, but as stated earlier, doing research on an ongoing project is an interesting point in itself and gives the opportunity to study the project in practice as it happens.

4.4 Access to the Project Organization

When doing a case study or any kind of research in general, it is important to have the ability to get access to the relevant data and sources. As a student at NHH, as well as having been employed part-time as student teacher, I already have an idea of how the organization is structured and some of its procedures. It has probably also given me an advantages in the way that it was fairly easy for me to be allowed to perform the interviews I did, and the project manager and other project members would happily suggest other people I should have a talk with. Respondents both from NHH and Bouvet had no hesitations to giving me interviews, and it is my impression that they enjoyed talking with me as researcher about the project and their experiences.

I tried to request some project documents and presentation, which I was made aware of while doing the interviews, to be sent to me for further analysis and possible use in this thesis for examples and appendices. However I have unfortunately been unable to receive much of this information. Although I believe I have been able to get a proper view of how many of the processes in the project work, it would have been advantageous to have some documents to support some of the information revealed in the interviews, as it could have strengthen the reliability of my study.

I do understand that members of the project may not be available to participate in additional, voluntarily activities due to limited time and resources, and I highly appreciate the time and resources which were made available to me.

5. Analysis and Thoughts on the Portal Project

This chapter will describe my findings on positive effects and challenges as a result of using a Scrum approach to use agile software development on the new portal. I will look at how the findings correspond to the theories, and whether the experiences from the portal project substantiate the theories, making them more or less reliable. All together this analysis will allow summing up some conclusions in the next chapter.

5.1 Propositions

I have made a few propositions based on what I have learned about agile software development, project management in general, and the practices in the portal project:

1. Although not included in agile software development (in our case Scrum) the PSO perspective is important and must supplement the project or development method used in an IT project, if it will affect the people and the organization.

2. A project is not transparent just by tracking performance and progress and putting it in a computer system. The information must be made easily available and understandable for those who request it.

3. Agile software development and Scrum is a technical project management method which works well on coding, and thus the system development perspective of a project. The people and organization development perspective are non-technical and works well with milestones and traditional project management.

5.2 Scrum Light

NHH and Bouvet have agreed to apply Scrum as an agile development process in the portal project, but they have decided not to apply all the characteristics described in the theories concerning the method. The reasons for this are a limited budget as well as the project’s relative small size might make it inconvenient to use all roles and processes suggested by Scrum. It is also a more moderate approach since Scrum is quite different from traditional methods, and hopefully a light version will be easier for NHH to adapt to. Some project members have explained that Scrum is a good theory, but to follow it to the point is not practical for this project and one must be realistic and adjust it to suit NHH's project. However since much have been left out, NHH and Bouvet prefer to state that they use a "Scrum Light" method, where "Light" indicates that not all of Scrum is being applied to the portal project.

The most essential scrum principles that have been left out are that there is no scrum master, there has been little emphasis on the prioritization part, and the whole project team is not working together in the same location. However the most important elements (according to respondents) have been implemented in the project. Developers have daily meetings with NHH regarding development. They run 4-weeks iterative sprints, including sprint delivery and meetings for planning and review, and they use the concepts of both a product backlog and a sprint backlog.

The lack of a scrum master definitely has an impact on the portal project. It is a scrum master's responsibility to make sure the project is a scrum project, and that methods and techniques are adhered to throughout the project lifetime and organization, as well as to teach project members about the method. Although members have been taught about the method and there is no doubt that everyone I have talked to is aware of scrum and that this method is applied on the project, there appears to be a significant difference in members' understanding of the method and its benefits. It is Bouvet who are experienced with Scrum and who have suggested the use of the method, whereas NHH are new to the method and need to learn how it works. Thus it is Bouvet who has taken the scrum master's role of teaching the method and setting up tools and guidelines to support this way of working on the project. For instance they have taught about the daily meetings, sprints, reviews and planning, and also set up a SharePoint project site where information and tasks are available for everyone involved in the project.

However since there is no scrum master, there is no one who has the continuous job of making sure the project is a Scrum project and making sure everyone completely understands how and why the scrum method is being used. It may also lead to some parts of the project not living up to what is initially required by Scrum. This might be the reason why there is still skepticism and uncertainties about the method among several of the interviewed project members. Much of the skepticism appears to be due to not knowing exactly what is being worked upon in the project or what the current progress of the project is, this indicates that much information is not available to all project members, or at least not easily accessible. This contradicts both with Scrum and Agile Software Development, which strongly requires information and progress to be highly transparent. The interviews I have conducted give me the impression that such a transparency throughout the project organization is not the reality. On the other hand there seems to be high transparency and understanding between the developers and the portal sub-project manager at NHH. This is no surprise since these meet on a daily basis through the scrum meetings.

Secondly we also have the lack of emphasis on the prioritization part. This means that in the sprint planning meetings prioritization has not been of such importance as it should be. However, this is mainly due to the project's state, and that it is currently only a basis for the portal that is being built; and as one respondent said "it is limited how creative one can be when developing a basis." It is simply just something that needs to be done, and which takes some time before we can start being creative and suggest new ideas to be prioritized for development. It seems that this is something which will change when the project enters the second phase, i.e. the portal basis is done and ready for adding new features and tools. At least it should change, if not a very important characteristic of Scrum is being left out. There already exists confusion with priorities as project members do not easily have access to a list of all suggested features, their prioritization, and current status.

Scrum requires developers to work close with their customers, and the agile principles states that "business people and developers must work together daily throughout the project" (Agile Alliance 2001). Apart from the daily Scrum meetings, this is not the case in the portal project where the developers from Bouvet are located in Stavanger and the rest of the project team in Bergen. Bouvet had its best SharePoint developers in Stavanger, and NHH decided that having the most experienced people was most important. If the Scrum methodology would have been followed to the point, Bouvet's programmers should have been relocated to NHH working together with the rest of the project team. Due to a tight budget this is not the case. In addition, having to keep transportation costs low also puts limits to the degree of developers' presence on site at NHH.

There is also a positive aspect of having developers located in their own company offices, they are close to their colleagues and can easily get assistance and support from them when needed. But it is with good reasons that having developers working together with the rest of the project team is preferable, and the portal project experiences negative effects due to this. At NHH some project members have admitted that they miss being able to sit next to a developer and point at the screen in order to explain or present thoughts, ideas, and suggestions. Although digital communication methods has come far and become quite advanced, it does not replace this kind of social interaction and thus the physical distance between Bouvet's developers and NHH limits communication. Bouvet's developers also notice the lack of communication when everyone is not located together, and indicates that this might be the most visible difference on a daily basis between Scrum and Scrum Light.

The daily scrum meetings, the sprints, review and planning meetings, and deliveries are conducted quite in accordance with the Scrum theory. The portal project has been strict on both having 4-weeks sprints and not skipping any daily scrum meetings. Each sprint starts with a sprint planning and a plan for what to be done and opportunities for NHH to prioritize and come with suggestion (to the extent developing a basis allows creativity). Further every sprint delivers new improvements and an enhanced prototype of the portal in the end, as well as allowing for a sprint review meeting. The daily scrum meetings are conducted between the developers and NHH's sub-project manager for the portal part. These are short 15 minutes meetings occurring at the beginning of the day every morning. Some meetings might last longer if necessary, but on a general basis both NHH and Bouvet claim that they manage to keep these meetings as short as they are supposed to according to Scrum.

5.3 Composition of Project Members

Analyzing the project organization one realizes that there are quite a lot of people involved in it. It gives an impression that maybe there are too many involved relative to the size of the project, possibly contradicting Agile development which supports small teams. However, very few are working on the project full time, and many are simply just involved in order to make sure everyone's opinions are being heard and representing various interests at NHH, such as the steering group and the project support team. Those working on the project on a daily full-time basis form a smaller team which is able to collaborate efficiently.

It is an advantage that NHH use people from their IT department which has the highest technical knowledge at NHH to cooperate with the developers at Bouvet. It is well known that business people and engineers speak a different language. Although members from the IT department have varied backgrounds, their IT knowledge is high and this should ease communication with Bouvet. Yet non-technical opinions must be heard, and this is where the steering group and support team comes in. Their contribution to the project is of high value in order for it to not be too technically oriented. The portal project has successfully managed to gather and include people from a range of interest groups at NHH as well as with different backgrounds. There are members representing students, staff, administration, as well as researchers. The sub-project manager for content is from NHH's information department and thus naturally having a good overview of information and flow of information at the institution, a knowledge of great value when be responsible for how information is to be presented in the new portal.

Except for the one person with former IT project management experience (including other forms of iterative development), there are few people from NHH with considerable project work experience and no one with former experience from Scrum. This provides a challenge for the project organization as well as for Bouvet to educated participants on the Scrum method. As already mentioned the lack of a scrum master creates an additional challenge related to providing teaching and understanding. Most likely this will decrease the efficiency of the project work, at least in the beginning, but on the positive side it provides an opportunity, especially for NHH's IT department to acquire knowledge and experience with Scrum, which can prove quite valuable in future projects.

5.4 Collaboration

The Agile Manifesto emphasize on "customer collaboration over contract negotiation." The importance here is the collaboration between the contractor (Bouvet) and the customer (NHH). NHH does have a contract with Bouvet, since clearly some kind of contract must exist in order to define some responsibilities. However, it is not a detailed contract explaining exactly what the end product is going to look like, including a list of all necessary features. Instead NHH and Bouvet have agreed upon the application of Scrum (or Scrum Light) and what this implies with regards to close collaboration and flexibility in terms of adding, changing, and re-prioritization at any time before the project ends.

NHH is generally satisfied working with Bouvet, indicating that they are professional and to the point, preferring to get things done, which is highly appreciated by NHH. This is perhaps a result of using Scrum. It is important to know each other well when going to work together on the project simply in order to understand each other and generate trust.

One respondent said that generally contractors often get too “house warm” and spend too much time on small talk rather than work. This has however not been the case with Bouvet, and they seem to be professional in the way they work. This satisfaction is generally regarding NHH’s experience with cooperating with Bouvet, and must not be mixed with the fact that there is also some dissatisfaction regarding availability of information.

The daily Scrum meetings and Bouvet’s project manager’s presence at NHH is a major factor for ensuring continuous collaboration. Thus it is reasonable to believe that Scrum has a positive effect on collaboration in the way Agile development intends.

5.5 Handling Tasks in the Project

Based on the results of my interviews, I have found there to be two categories of tasks in the portal project, one being task related to the technical development, or programming, or the portal, and the other being all other tasks that must be done, such as gathering information, planning meetings, making decisions, etc. The development tasks are primarily performed by Bouvet, whereas all other tasks are performed mainly by NHH, although some also by Bouvet (and possibly also KPMG and Inspera later).

5.5.1 Entering Tasks in the Backlogs

As mentioned earlier, the portal project uses a backlog in their light version of Scrum. In fact they use both the product backlog and the sprint backlog. NHH decides the items in the product backlog and prioritizes the list according to their preferences and then presents it to Bouvet in the sprint planning meetings. The developers at Bouvet then cut each task into smaller tasks requiring preferably about four works hours, and never more than eight – this is the more technical sprint backlog. It is the sprint backlog which constitutes the list of technical (programming) tasks to be done. Due to the flexible nature of Scrum we understand that this list will also be subject to change on short notice as a result of changes in the product backlog. Other tasks are listed in the project portal, and some are just given or done ad hoc.

Whenever NHH has a new idea or a feature request which is brought to the developers, they (the developers) enter it into a computer software application designed to keep track of tasks and their status. This is a way of making sure that all requests are organized and easily accessible. It is the list in this computer system which in practice constitutes the actual product backlog, and it is in the same application that one breaks each product backlog item down into one or several sprint backlog items. The developers keep this list as a list of work to be done, and it is NHH who prioritize what should be done first. There are of course some exceptions to this now that the project is in such an early phase. Now most of the development is related to building the foundation or basis of the portal, which just simply requires some work to be done, which is out of NHH's knowledge, and thus is planned and prioritized by Bouvet, but features that can be modified are of course informed about to NHH. In practice, the engineers know what need to be done in order to finish the foundation of the portal and works on this, but when decisions regarding functionality have to be made they inform NHH about this. Say for instance when they have reached a stage where they can start setting up the graphical user interface, NHH is informed about what is possible to do, and NHH gives their preferences.

For the technical tasks, the engineers claim that Scrum breaks away complexity by the way it forces one to make short tasks lasting no more than eight hours. It is easier to work on a small task than a big and complex task which it is hard to get an overview of. After having started using scrum in their development projects, the Bouvet engineers have little doubts that they prefer Scrum to other development methods.

One project member shows high satisfaction regarding short and simple tasks: "Scrum gives a short time frame, with tasks to be solved in short time. This is a simple way to work, and motivates me." This shows that Scrum actually makes simple tasks motivate employees, and it is clear that motivated employees can have nothing but a positive effect on the project's performance.

5.5.2 Prioritization with Post-It Notes

It has been suggested that one should use a board with post-it notes of different colors when prioritizing features in the product backlog. Each post-it note will then represent an item in the backlog, and the board will give an overview when the list is to be prioritized. It is easy to make priorities by moving the post-it notes in different places depending on how one wish to prioritize the different tasks. The project manager say they wish to have some sort of a post-it board in the portal project as well, but it has proved difficult since the developers are not located in Bergen. It has been mentioned that one might be able to find a digital solution to solve this problem, however so far they do not have a tool for this.

5.5.3 Short-Term Planning

Scrum tasks are performed in the sprints, i.e. in iterations. Before each new sprint one plans what do be done in the next sprint. And when done with the meeting the next sprint starts, and hopefully what has been planned will be done. The planning is very short, in the way that one only plans what will be done one month ahead. One project members say that "we know very much about the current sprint we are in, but little about what will be going on in the next one." The member further explains a wish for some sort of overview, a plan for what will be done in the whole project. This is a difficult issue, because the whole idea of agile software development is that one is not going to plan far ahead as it can limit the flexibility of the project. One can ask: do you really need to know what is going to be done in the next sprint as long as you know everything that needs to be done in this sprint? And also know that you will know everything that is going to be done in the next sprint anyway, but not until just before it starts. Why worry about the next sprint during the current sprint? I have yet to have found a good reason why one needs this information, except maybe for curiosity, however if project members are uncomfortable not knowing anything about the next sprint, this is an issue that must be addressed.

The portal project does indeed know a little more than just what is going on the current sprint, as they have divided the project into phases, and know that phase one will be complete in June 2009, and know that it's completion is defined by a fully functional (although limited on functionality) portal that is ready to be introduced to staff, students, and researchers. However, except for this, project members do not know more about the future than what is planned in the current sprint. If we are going to trust Scrum and agile software development, we shall not plan far ahead, and in the case of Scrum, it appears to me unnecessary to know much about the next sprint. It is my understanding that knowledge about the next sprint should not influence and cannot improve the performance of the work performed in the current sprint. Let me illustrate with an example:

It is of no value for a postman to know tomorrow's route for mail delivery, as long as he knows where to deliver mail today and tomorrow's route will be available when he needs it, i.e. tomorrow. I am sure a postman would not care much if he did not know tomorrows route, all he is going to do is to deliver mail anyway according to the plan. A postman's job does not require critical thinking regarding the route and as long as mail is being delivered everything is fine. (I admit not knowing much about postmen's job, but use this only as an illustration, and under the assumption that a postman's job is to deliver mail only, according to routes prescribed by his superiors.)

However, say if we were to put one of NHH's researchers to deliver mail one day, he might start questioning whether or not the route was the most efficient way of delivering the mail, wanting to have a look at the plan for tomorrow in order to influence it, and being satisfied with his route having been planned sufficiently before doing it. A researcher has been trained to be a critical thinker and to gather as much information as possible. Now if we were to apply this illustration and assumption to the portal project, I believe we might have a similar situation. It is my own personal impression that we who study business are trained to seek information and be critical thinkers in order to finds ways to do things better and more efficient, this will of course account for researchers and possibly other staff as well. Participating in a Scrum project which is quite different from the way we have learned to work in project, it might be quite uncomfortable not knowing things that one is used to knowing in a traditional project. Perhaps this is the reason why some project members are uncomfortable not having enough information about what happens after the current sprint. There might be a feeling of lack of control? If this is the case it is understandably more comfortable since this is the first Scrum project, and one will possibly get used to this being the way a Scrum project work, and if the project turn out successful using this method, one might be less resistant to not knowing what is going on in the next sprint in the next Scrum project.

5.6 Measuring Project Progress

In Scrum the backlog serves as an indicator of the project's current progression, and also allows us to predict – through the burndown chart – when the project will finish, or how much is expected to be accomplished by project termination (Shojaee 2008). Each task, both in the product backlog and sprint backlog is given an estimate on required hours of work. The estimated numbers of hours for a product backlog task is of course equal to the sum of all sprint backlog tasks which relates to the given product backlog task.

5.6.1 Storing Information on Progress

The developers at Bouvet keep very well track of the progress information in their developing computer systems. As described earlier, they list all the ideas, feature requests, and tasks in their database with current status and time estimates, and they have them available in both the product backlog, as well as in the more detailed sprint backlog. I have seen that this information is stored in the database. The information is not necessarily as organized as one would expect it to be from the theoretical point, i.e. tasks are not performed exactly in the prioritized order, some tasks are unclear, and some tasks lacks full descriptions, but in practice this is not a serious problem, and a complete, up-to-date, and “perfect” list would most likely prove inefficient as the programmers should have their primary focus on developing, and not updating a list (quite in line with agile software development’s strong focus on development). One does not stop three times in the middle of a 4-hour task to update that it is 25, 50, and 75 percent done. The information is there and it can provide measurement of progress.

5.6.2 Retrieval of Progress Information

Information on progress is stored in the developers’ computer systems, and can also be retrieved from the same system. The engineers can see which tasks have been completed, which are under progress, and which are to be done later. They can see how many hours that are estimated for completing each of the remaining tasks in a current sprint, as well as to see both the predicted and actual number of work hours for tasks already completed. According to Agile software development progress is to be measured by working code because working software is what the customer wants and needs. The portal project measure its progress by the status of the task or list of things to be done, which is in practice what has been coded or not, and thus quite in accordance with the theory.

The developers were about to launch the very first working demo as the delivery for the current sprint in the week after our interview. Knowing that one had already come this far, and also knowing how many sprints were left, the project manager could state that he was certain that a working portal (with limited functionalities) would be available as planned by the end of phase one. It is my impression, however, that this is mainly a feeling based on what has been accomplished so far rather than actually having sat down an calculated how far one would have come by end of phase one and how many functions would be available. By simply knowing how many work hours were left in the project before end of phase one, one should be able to use this information and compare it with the time required to complete the requested features, and have very detailed information on how much is actually going to be delivered by the end of phase one. If one knows that only features A, B, C, and D can be completed within the timeframe, one should be able to state that by the end of phase one we will deliver a working portal with functions A, B, C, and D. Functions E, F, G, H, etc. will be worked on in phase two, and delivered continuously. I was unable to get such a specific answer, but it was certain that the basis of the portal would be ready, and regarding functionality "as much as possible" would be available.

5.6.3 Transparency Issues

Regarding retrieval of progress and information on the project in general, there appears to be some challenges regarding transparency, availability, or ease of access to information. Some project members have described wishes of lists of information that I have found already existing while talking to others. For example one project member wished there was a list of tasks that had already been accomplished, and not only a list of things to be done. This information is already available in the developers' computer system, and should also be accessible by all other project members.

I have found there to be several possible reasons for this. The first is that all project members probably do not have direct access to the computer system which the engineers are working on and storing this information, since this is located in Bouvet's office, and if the information is not transferred to a system where all project members have access, then there is indeed much valuable information which unfortunately is not available to the whole project team.

Secondly, and if all do have access to the database in which information on tasks are stored, I believe it might be too technical for non-technical project members to understand or use. The user interface of the system is quite complex with a lot of functionality and many ways of presenting the information (by priority, by status, by sprint, only product backlog items, or only sprint backlog items, etc.). When asked, the developers can present most of the information I ask for, such as product backlog, sprint backlog, priority of items, estimated and actual number of hours, and status of each task, but I get an impression that one needs to know how to use the system more than one can expect a typical non-technical member of the project to be able to without training. The system shows the information in list of a number of items, which can appear unorganized if one does not know how to retrieve the information. It does not present the information in a simpler or more visual way such as in graphs or a burn down chart. The PM at Bouvet does however make graphical presentation based on these number, and present to NHH, however, I was unfortunately unable to view such a presentation on short notice, which also indicated that it might not be so easily available.

Thirdly, if all this information is available to everyone involved in the project, there must have been lack in informing about this or training on how to get this information. This is unfortunate simply because project members do not know how to get access to this kind or information, and do not know they have access, and thus are frustrated over something which could easily have been avoided.

Scrum and Agile software development advise close collaboration and working together on the same place because the best way to convey information is by face-to-face conversations. If everyone had been in the same location the transparency and information issues might have been more easily avoided. The sub-project manager for the portal which is in daily meetings (let’s call them “almost face-to-face”) with the developers does however not seem to have any lack of information, which supports using face-to-face meetings rather than written information such as reports or e-mails, etc. The challenge, for Scrum and Agile development, would however be how to make sure information goes out to everyone, because all conversations which go on in a project cannot include everyone in the project. So when information is conveyed through face-to-face conversation, those not participating in the conversation will not receive the information. One must be aware of this and find other ways to ensure necessary information is made available and passed on to others.

5.6.4 Missing Milestones

As a result of not knowing much about what will be going on in the next sprint, or what the project is moving towards, it has been stated that it is difficult to get an overall overview of the project and its progress. It might be due to being uncomfortable adjusting to a new project management method; however it is still being suggested that the project should have some milestones. It need not be detailed milestone as in a traditional project, but rather some overall milestones that show how far the project has gotten, and what is left. I find this rather difficult since Scrum is very much about not planning any kind of path in advance, simply in order to allow as much flexibility as possible. Working with Scrum you will not know exactly where the project is going, and thus it can be difficult to plan milestones much further ahead than past the next sprint anyway. However, I am informed that although there are many uncertainties in the project there are still some features and task which one knows the project must include, and which need to be done in a certain order and at specific times. Such as conducting the pre-study, launching the portal, offering trainings, performing evaluations, and so on. These could have been put into a milestone chart or a GANTT diagram.

After further investigation it appears that many of the items which could be put in a milestone chart are things not related very much to the daily development of the portal, the system perspective, but rather the people and organization perspective. Indeed tasks related to the people and the organization are not put into the product backlog, but rather in the task list in the project portal. Since Scrum measures progress based on the backlogs, it is evident that it is only progress in the system part which is measured; progress on the people and the organization perspective is actually left out. Thus perhaps we might be in lack of a PSO perspective when it comes to measuring progress. Perhaps we should try to combine Scrum with milestones, by taking into consideration Andersen et al.'s (2004) suggestion of three paths in Goal Directed Project Management, one for each of the perspectives: (1) people development, (2) system development, and (3) organization development. The path for system development is the Scrum path, and can be left pretty much free of milestones, in compliance with the theory, but perhaps only including the phases and/or sprints.

Merging Scrum and the use of milestone in this way could prove valuable in order to merge both technical project management and managerial project management. I will try to suggest an illustration in figure 8:

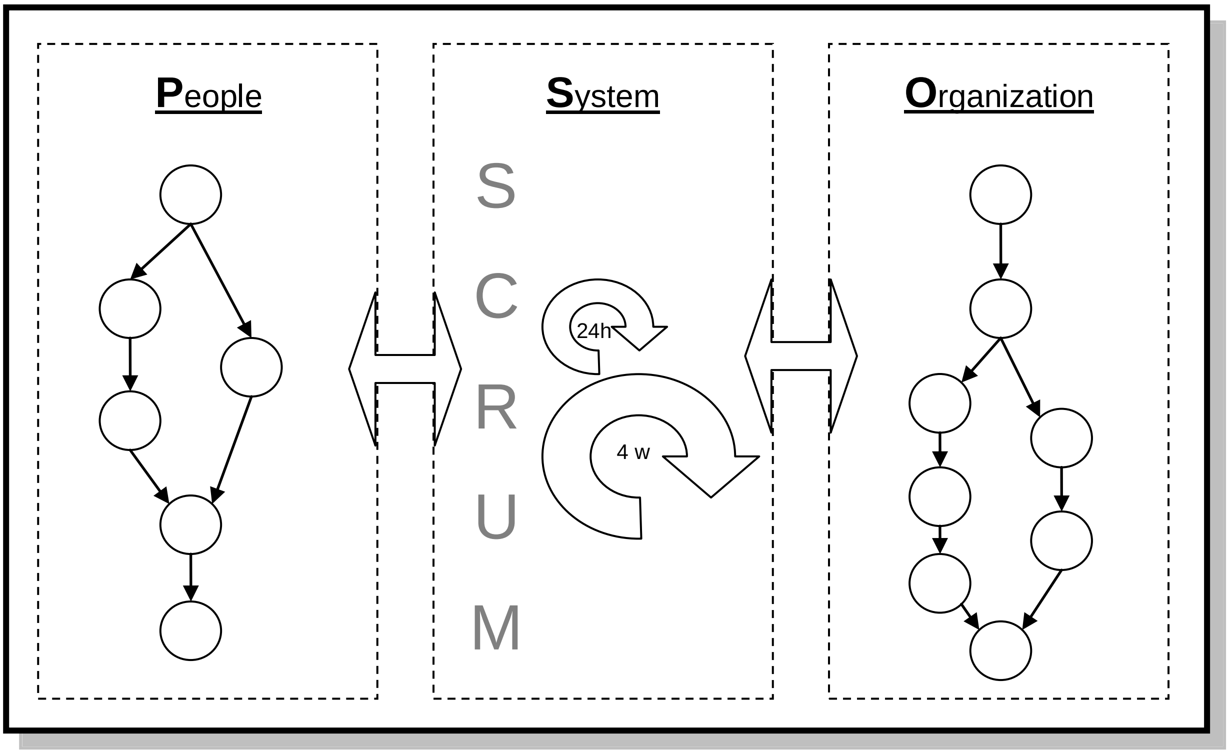

Figure 8: Combining Scrum, Milestones, and PSO

This figure illustrates how one can use milestones on the people and organization development paths of the project, while still letting system development follow the Scrum methodology. One can of course find way of letting milestones from people and organization to depend on each other as well, and even to some extent depend on have reached certain stages in the scrum development too, e.g. the trainings will not be conducted until after system development have released a working portal (phase 1), however awareness among students, researchers, and staff must have been created before this point. I believe this combination of Scrum, milestones, and PSO will allow the technical aspects of the project to be managed in a technical way, whereas the non-technical aspects are managed according to more traditional project management theories.

5.7 Making Decisions in the Project

According to Scrum it is the product owner who presents and have the main responsibility of prioritizing the product backlog before each sprint, however, the person defined as product owner in the portal project does not have a product owner role as defined in Scrum. The project owner’s role in the portal project is simply being a member of the steering committee as a representative of NHH’s management and administrative board which provide funding for the project. The portal project claims to having received enough authority to make those decisions which need to be made, however there is not one single person with responsibility of making decisions alone. Decisions are made by getting opinions from those involved in the decisions to be made, such as representatives for various groups. If the student representative presents some student preferences on what information they would like to have in the portal, then there is no reason for anyone not being a student overriding that preference, because the look of the students' version of the portal will affect students only, and thus it is the students' opinions which are relevant. Prioritization of features and task are done in collaboration and weighed by cost and value. In general it does not seem like the way decisions are to be made have been pre-defined. It seems like the product owner, by Scrum's definition consist of all who has an opinion on whatever topic discussed.

This way of making decisions appears to work very well, and I have not been notified by anyone at NHH of any dissatisfaction with this. It is important that people that will be affected by the project are involved in the decision making, which is also in accordance with applying a PSO perspective. Yet NHH is a public institution, and public institutions are not well known for making decisions fast and efficient, but rather the contrary. This can be a problem because if the developers need NHH to make a decision it will take some time for it to be made since several people must discuss it, instead of one single project owner replying more or less immediately. The developers at Bouvet confirm that this is the case in the portal project, and it does take longer to make decisions working with public customers than when working with private customers. They work around this by continuing on the next task if one task must be put on hold due to having to wait for a decision. "We cannot just sit here and wait and do nothing", thus some functionality might be developed not quite in the order of the prioritized backlog. This is also the reason why some backlog items further down on the list are completed although there are still some above which are not.

Although not quite following the rules of Scrum, it seems that the portal project handles decisions well. There must be a trade-off between making fast decision and making decisions which everyone is included in and feels ownership to. It might make decisions slower than preferable, but it is no major disadvantage that tasks are not performed exactly as prioritized in the backlog to compensate for tasks put on hold.

As a result of the way information is stored, and that one does not write minutes of meetings, it can be hard to track what was decided upon at the former meetings, or get a list over things that were agreed upon. One project member say that “I think it is difficult to find out what was decided, and when.” The items on the product backlog should display what has been decided, or the tasks in the project portal, however neither includes any information on when the decision was made. It must be up to the project to decide whether or not it is important to be able to read reports from meeting, if so, then they should start writing minutes. I do however believe that it is the general transparency issue which is the main problem, and if solved, there will probably not be a need for meeting minutes. The only issue here is that the product backlog will not include features that one has decided not to develop at all, i.e. features that are not prioritized far down, but rather not put into the backlog in the first place, or having been completely removed later. There is no list of features one has decided not to implement, perhaps this could prove necessary?

5.8 Degree of Agility and Flexibility

Since the portal project uses Scrum sprints, and conducts the sprint planning meetings before each sprint together with the accompanying product backlog the project is definitely agile, and the developers are flexible in terms of what is being developed and in which order. There is no doubt that the project method invites NHH’s IT department to participate more in the development than a traditional waterfall model would, and they do take advantage of this opportunity to influence.

The portal project is a creative project, in the way that they are making a completely new operational portal, which has never been done before. This requires a high degree of creativity so that they can come up with what kind of portal this is going to be, what it can do, how it will look like, what kind of needs it will cover, and so on. Some respondents have commented that it is due to the need for creativity that Scrum is well suited for this project. Although one can argue all projects need to be creative if they are to be defined as unique tasks which have never been done before, I will assume that it is the size of the project which requires *more* creativity than previous much smaller projects have ever done before at NHH, and that is the reason for relating Scrum to creativity.

The agile and flexible nature of Scrum allows NHH to be creative throughout the project, and not just in the beginning of the project. If they get any new ideas as they see the portal being developed, these ideas will be taken into consideration, and implemented if NHH give them high enough priority in the backlog. This is an important advantage of Scrum, and the portal project is to some degree utilizing this advantage. The reason why I state “to some degree” is, once again, due to the fact that the project is currently in the state of developing the foundation of the portal, and that this has not allowed too much creativity so far in the project. NHH has still had significant influence on the project so far and probably much more than if using another development method than Scrum, but the stage of the project still limit the degree of flexibility somewhat. The developers cannot wait with developing the foundation, it must be done first, and thus NHH experiences less flexibility now than they will once the first phase is completed. After the first phase, most, if not all, development will be feature requests from NHH, and then NHH will probably get a stronger feeling of their controlling the development than what the situation is today.

One project member at NHH states that when it comes to flexibility it is mostly about flexibility for the developers. This goes against everything that Scrum and Agile software development stand for, nor is it the interest of the developers at Bouvet. Thus this feeling is most likely a result of the issue described in the previous paragraph. The iterations in agile development and the sprints in Scrum are there in order to be flexible for the customer. It is in a waterfall-model where the developers go away for a year and program alone until all predefined requirements have been met, that the developers are flexible to do whatever they want during development. This should not, and cannot be the case while using iterations and sprints. The feeling of lack of flexibility can also be a result of the transparency issues with information, since everyone does not have the same information with regards to the status of the development, and thus generally has a feeling of not knowing exactly who is in control of the project.

Throughout several of the interviews at NHH it has also been a tendency to mention the budget, and that the budget limits the size and the potential of the project. This does not limit NHH’s project members’ ability to influence the development or prioritize features because the developers are flexible enough to meet these changes. However it does put limits to the degree one can suggest ideas because each idea, even though it might be considered a very valuable feature, must the evaluated towards the budget and its cost to be developed. In this sense the project itself is less flexible because some expensive ideas might have to be put on hold until one know if there will be cash to support it. We know that currently the whole process part of the project has been put on hold, and thus many ideas or possibilities related to integrating solutions for processes in the portal will not be considered or developed.

5.9 Focusing on the Budget

The portal project must stick to a strict budget, and all development must be within the amount they have been given for the first phase. It can be a challenge to combine the strict budget with Scrum, because Scrum does not seem to easily adjust to the way they need to adjust to budgeting at NHH. Further the project's development is planned only for one month (sprint) at a time, and subject to change at any time.

Regarding budgeting it is naturally very difficult to make estimates on how much time a task is going to take, when one is developing something that has not been done before, and thus the estimates made and used in the portal project are very rough according to the Project Manager. "Our estimates are so rough that we are certain that we will miss, and do not care much whether or not we hit the estimates." However it will allow learning, and thus estimates later in the project will perform better. But it seems from NHH's perspective that Scrum is not designed to allow budgeting for this kind of estimation. NHH seems uncertain in what ways other Scrum projects follow budgets, as Scrum appear to makes decision only based on whether or not a new idea is perceived as giving a positive effect or not. If it is perceived positive everyone agree to put it in the backlog and just do it. The portal project has to make sure they do not exceed their budget. The Project Manager proposes a so-called "virtual pool" of work hours to go together with the rough estimates in the project. In this pool one could deposit and withdraw work hours. Say if one tasks takes three hours shorter than predicted, then these hours would be deposited in the pool, and available for use for other tasks taking longer time than estimated. "This is not how Scrum thinks, but I think it could have been an advantage."

It is understandable that in a public budget-focused institution that does not operate based on generating income, but on sticking to a budget, this is important in the project. This kind of information could be interesting to have as well, but it should in theory not be of much relevance for the project's progress when Scrum is followed. If we assume that Bouvet and the developers simply develop from the product backlog starting from the top and working down as far as possible within each sprint, it should not really matter how many hours more or less they use for each task, because one just continues down the list anyway, and all available work hours will be used. It is true that Scrum does not look at the amount of money available and plan developing based on that since that is more traditional development. Scrum simply starts development right away and continues developing until the budget is empty. Once no more resources are available, the project is terminated and the product released has the most important features that were possible to develop based on the budget available.

I cannot see why planning based on budget would change this, because due to the nature of the product backlog it is always the most important feature, not yet completed, which is currently worked upon until the project ends. Skipping these unnecessary planning hours will simply only make more hours available for development. Say you have three features, A, B, and C prioritized in the same order, but only A and B can be developed under the current budget. If we use Scrum, A will be developed first and then B, and perhaps during development of C one will run out of money, and the product released will only include A and B. Is there any way the product would include any other combinations of the features if one knew beforehand that only two features could be developed? We already ranked A and B as the two most important, so even if we knew all three could not have been developed due to budgeting and estimation beforehand, there is no reason for choosing to develop only A and C or only B and C, it would still be only A and B which would be developed, because they are more important than C.

5.10 Applying a PSO Perspective

The theories regarding the need for a PSO perspective has a tendency to focus too much on the technical aspects of the project. For the portal project, this means there might be a chance of focusing too much of the technical development and programming of the system, rather than including students, researchers, and other employees affected by the new portal. However I have found that the portal project is indeed very successful in adapting a PSO perspective, both in the way project members are thinking, and in the way the organization itself is structured.

5.10.1 Informing and Creating Awareness

The portal project has through a series of articles in NHH’s staff’s internal newspaper (Paraplyen 2008, Marthinsen 2008a, 2008b, 2009) presented the project, its status, and purpose. In addition they have even published their own blog (www.nhh-vp.no) dedicated solely to the portal project. I believe that such a blog could have been a terrific way of keeping everyone interested in the project updated on it constantly throughout development by simply clicking in and reading the latest project news. Unfortunately the blog has not been updated since the first post in October 2008. The sub-project manager for content, which is responsible for the blog, has had to prioritize it down due to the need of more important tasks related the portal. It seems however that the staff and professors at NHH have been in general well informed about the project. As a student, however, I cannot personally say that we have been informed to the same extent, and information to us could have been improved. The sub-project manager of content is aware of this, and says they are currently working on getting an article in K7 Bulletin, the students' newspaper. Hopefully this will increase awareness among students.

5.10.2 Including NHH People in the Project

It is the users that will be using the new portal, and the users include the following groups: students, researchers, and administrative staff. The portal project acknowledges that they cannot just make a product and then simply make the users use it. Instead they must include the people who will be using the product in the project, and allow them to have significant influence on development and the final product. It is the portal project itself who has defined these user groups, and that itself shows awareness of different users having different needs. Segmenting them into group will easier allow customization for each of the groups as well as to more easily identify who one needs to talk with when developing the portal. "We want them to think together with us."

As we have seen, the project organization is organized with a steering group, which also has members who can function as representatives knowing the different needs for each of the user groups. Further the sub-project manager for content has an important part as he has the overall responsibility for what information will be displayed in the portal, and most importantly how it will be displayed differently for the different user groups. To cope with this he will constantly talk with representatives from each user group and ask what their opinions are on different topics. These representatives are not randomly selected, but chosen based on their position in their group as being people who naturally will be able to represent a large part of their group. "It would not make much sense to randomly select a staff member, and end up with a janitor representing what tools staff needs in the new portal. We select people who we know will be using the portal to a considerable extent, because they will have more valuable opinions" says the sub-PM for content.

These representatives will give their opinions on features to be implemented in the portal, but are also allowed to bring new ideas which might prove useful in the portal. After having conducted a meeting with the user group representatives the sub-PM will note down all the opinions and ideas which have been presented at the meeting and work further on how these can be made into reality in the portal. This does for example include making draft of how the portal should look like based on the information received at the meeting. After this is done there will be a new meeting where the representatives can look at the graphical drafts and give feedback on whether or not the sub-PM has understood their thoughts. Once able to see the ideas graphically there might be new feedback such as "that was not quite like we were thinking", "maybe we should put that box further down on the page", or "can we do this or that instead?"

Conducting these meetings might take much time and effort, but it can prove to be of high importance to the success of the project. The portal project wants the users to use the portal when it is launched, and if they are to use the portal it must serve their needs. The project learns about these needs by listening to the users. In addition, by feeling involved in the project the users will get ownership to the decisions taken as well as to have more understanding for the reasons behind those decisions. This increases decisions acceptance and will lead to less complaints later. The way the portal project includes the future users of the portal does to a large degree live up to ideas behind a PSO perspective, and will most likely prove to pay off in terms of value for the users when the portal is made available to them.

5.10.3 Offering Trainings to the Users

Another aspect of the project which proves a PSO perspective is the planned trainings to be offered for the new users of the new portal. Although these training are not actually being planned at this point in time and one does not know the details of how they will be conducted, the fact that one is aware of the necessity of performing these training, shows that there is a constant awareness of the future users of the portal.

So far the thought is to conduct trainings based on individual's IT knowledge as well as general feeling towards a new portal. There will be four groups of trainings: (1) training for those in support of the new portal, (2) training for those who consider themselves competent computer users, and (3) training for older and less competent users. The fourth and last training will be for those who are skeptical to using the new portal. This training will argue reasons why the new portal is better than old solution, and give recommendations and reasons why they should change their habits and convert to be users of the new portal. Knowing that there will often be resistance when it comes to changing habits and trying something new, one hopes that this training might prove to decrease the portion of people unwilling to use the new portal.

The training will be offered voluntarily, and no one will be forced to attend them, but hopefully attention around the new portal will increase awareness of the new portal and thus also interest in learning more about it. They will be available for all, not only researchers and administrative staff, but also students. Most likely there will be no problems having people sign up for the first three trainings mentioned, but the last one for the skeptics might prove a bit challenging. How do you get people to attend a training which goal is to change your mind? That being asked, it is still a good sign that one plan to conduct all these trainings, and it proves that one is aware of weaknesses that have been in many IT projects in the past, and which is one of the reasons for recommending the PSO perspective. It does not matter how good the new system is if you do not teach the users how to use it and how it can be taken advantage of. NHH does not only plan to teach the users how to use it, they have even segmented the users in order to give customized training. I believe the portal project’s PSO perspective is more than sufficient, and will prove successful when the portal is made available for the users.

5.11 Resistance to Change

There are two possibilities of resistance to change which are relevant for the project. The first being resistance within the project organization and the other being resistance from the people which will have to adapt to using the new portal when it is finished.

5.11.1 Resistance within the Project Organization

Resistance to change within the project organization can occur because of the adoption of a Scrum development method, which is new and different from traditional project management. While conducting my interviews I have experienced a general feeling from the respondents that they are indeed positive to using Scrum, and everyone can give positive arguments for why they are using Scrum, thus it seems like there is no resistance to using this new method. That is of course an advantage to start with. However there does exists some dissatisfaction with some of the implications of using Scrum, such as the lack of an overview or milestones and not knowing much about what one will be working on in the next sprint and later.

The Project Manager acknowledges says the following about getting people to adjust to a Scrum methodology: “Those who are typical waterfall people will have challenges adopting this method … If it is important for you to know where you are going – or where the goal is – then you will have challenges.” It is evident that the use of Scrum has led to frustrations because of these reasons, but the fact that most project members are still very positive to the methodology, indicates that these frustrations are mainly a result of adapting to something new, but which one will be more comfortable with after some time.

5.11.2 Resistance From the New Users

Resistance from users is likely to come because students, researchers, and administrative staff at NHH will have to change from the way they use NHH’s electronic services today to a new and currently unknown system. This is something one is sure will be the case, mainly because there are some older professors which has been reluctant to acquiring new technology in the past, and there is no reason why this would not be the case again. This is one of the main reasons why one of the trainings to be offered when the new portal is released will be aimed specifically toward this group of users, as an attempt to minimize the resistance experienced from those with a negative attitude towards the new portal, and perhaps technology in general. Resistance from students is expected to be low since they are young and used to adapt to new technology constantly. The administrative staff will most likely be positive as well, as long as thing works well and they are properly taught how to use it. The Student Administration has had very positive results of implementing the use of Student Web (Danilova and Danilov 2008).

Further it is also important to make sure that those without the technological resistance, but rather the resistance to changing habits and changing to something new are also taken into consideration. Members of the portal project understand that in order to successfully having users start using the new portal it must be intuitive, easy to understand, and better than the tools that are already available today. The trainings will teach these improvements as well as how to use it, but "if it is easy for people to still do things the old way, it is a big chance that they will continue doing things the old way." Because of this we will shut down some of the old tools which the new portal will replace, so that people are forced to try the new portal, and hopefully it will satisfy them to such a degree that they start using the other tools in the new portal as well.

5.12 NHH as an Academic and Public Institution

Project work at NHH is different than in a typical private company for two major reasons: First NHH is a public institution in Norway which primarily relies on funding from the Norwegian government. Secondly NHH is an academic institution among the top business schools in Norway, thus having many well-educated professors and other staffs in possession of high academic degrees.

Public institutions are not well known for being efficient and fast when it comes to work practices, and there may be many hierarchies and many stages for decisions to go through in order to be approved. Both project members at Bouvet and NHH realizes that decisions cannot be made as fast at NHH as they might have been in a private company. When comparing public customers with private customers, the developers agree that this is the case in general for their former projects as well. This statement is also supported by the project member with past experience as a project manager in a consultancy firm, adding that some of the projects they had with public companies or the government would spend much more time on making decision than private companies, as well as these decisions being subject to a lot of politics. The portal project claims to having received enough authority to make most decisions on their own, without management approvals. Still it is my understanding that decisions take more time because many people are allowed to offer their opinions and give feedback. This is partly a result of the PSO perspective and need for involving as many as possible of those who will be affected.

When it comes to being a top academic institution, there are many people who are interested in and follow the project. They will question the way things are done and the decisions that are being made. With a high degree of knowledge and years of research on business processes, management, economics, psychology, and so on, these people are equipped with a critical way of thinking which will lead to much more skepticism and difficult questions being asked than most other organizations or companies would. Members at NHH admit that they are probably a difficult customer for Bouvet when it comes to this, and the developers agree that NHH asks questions more than average. However this will be the natural behavior of an institution such as NHH, and it is important to allow the customer to ask critical questions, because it allows both NHH and Bouvet to test or think about whether or not they are doing things in the best ways. It also allows Bouvet to teach NHH more about the Scrum method. Some of the questions might also be due to curiosity about the new method from people who enjoy expanding their knowledge.

6. Conclusions

I started my thesis being curious about IT projects and the challenges in managing these. I formed four research questions as basis for my research, but I have found and learned many new things from the portal project which allowed me to present three propositions as well. I will emphasize once again that my findings are based on research on a relatively small project, which in addition is far from complete. However it has been an interesting study and I do believe my findings can apply to many other projects as well.

6.1 Answers to Research Questions

Does the portal project manage to adapt an agile software development methodology?

We have seen that NHH's portal project has adapted Scrum as a development methodology. The method has been adapted due to recommendation by Bouvet, and it is the programmers at Bouvet who work on the development daily. They have experience with this from before and know well how to conduct programming this way, as well as having proper IT systems to support this kind of development. Thus there are few obstacles for them to apply Scrum. NHH has however little former project experience, and considers this the largest project in the IT department in recent time. They do however generally have a positive attitude toward working on the portal project, and find the Scrum methodology exciting. Whatsoever there are challenges in terms of not having a traditional plan which shows a plan of the project from start until end, and some project members have found it difficult not knowing what will be happening in the project in the next month.

They have also been unable to fully experience the benefit of the product backlog and possibility to change and reprioritize this throughout the project since it is currently in such an early stage, i.e. developing the foundation of the portal which does unfortunately not allow much creativity. This is however expected to change drastically once a working portal has been launched and the next phase starts.

How does the portal project make agile software development work for them?

Instead of following Scrum theory to the point, the portal project has adjusted it somewhat and calls their method a Scrum Light methodology. This is a way of tailoring the method to suit the size of the project and budget constraints. NHH make the product backlog, and the developers generate the sprint backlog based on this. They work continuously on developing the portal, and meet with the portal sub-project manager daily, every morning for about 15 minutes. There is however no Scrum master, which result in there being not a specific person with responsibility of teaching the methods, and making sure everyone adheres to it, this, in addition to NHH having no past experience with Scrum, is most likely the reason why not everyone knows exactly what are the benefits of scrum and why certain things are not done in a traditional way.

In addition the portal project does not have a product owner in the way as described in Scrum, instead there are many people and representatives who need to be included in discussions before decision can be made. This makes decisions slower and appears to be both because NHH is a public institution as well as applying a PSO perspective, which requires including more people as part of the people and organization perspective.

How does the portal project handle the need for a PSO perspective?

Decision-wise it appears that many decisions are being made with considerable emphasis on a PSO perspective, and one is well aware of the project affecting the people and the organization. They take time to include representative for the users in the portal, and thus do their best to make sure that the portal is developed according to the users' needs. They realize that in order for the users to start using the portal, it must be a good product, better than the tools they use today. If this does not turn out to be the case, then the project will not be successful.

There exist however challenge relating to project members' overview of the project, apparently due to there being no overall plan or milestones (as a result of Scrum). It appears that Scrum works very well like this on the system development perspective of the project, but when it comes to people and organization development, I have found that it might be beneficial to use milestone here, and merge the use of milestone together with Scrum under a PSO framework.

What works well, and what is more challenging in the project, and why?

I am impressed with some of the excitement I have seen among some project members and that they enjoy working with this project and look forward to when results will be available in terms of a working portal. It is a major advantage to have people who feel motivated to work on the project. In addition it has been mentioned that short and simple task as a result of Scrum motivates people's involvement in the project.

A major challenge in the project is making sure that information is transparent, easily accessible and understandable. At least the product backlog and its status should be updated and available to everyone who is interested. If this kind of information is not available for everyone, and there is no specific reason for this, then it should be made available. If the case is that the information is already available, then members of the project should be taught how to find the information and understand it. There should perhaps also be more time spent on educating project members on the Scrum methods, what the advantages are, and why one has decided to use it. The reason why Scrum does not work as easy as it should right now, is due to running Scrum Light, as well as the project's being in such an early state. Members should be informed of the side-effects of running Scrum Light, without a Scrum master, and also regarding development that they are not able to fully experience the advantages of Scrum until the next phase when the foundation is already done. I believe this would remove some of the dissatisfaction with Scrum which currently exists.

6.2 Justifying Propositions

Although not included in agile software development (in our case Scrum) the PSO perspective is important and must supplement the project or development method used in an IT project, if it will affect the people and the organization.

We have seen that the portal project will affect many different users, and it will develop a system for the users, and thus the users must be able to bring their opinions forward when decisions are to be made. The PSO perspective is important, and has evolved as a result of IT projects focusing too much on the system development. Scrum does not have any framework to compensate for the PSO perspective, and in order to make sure one is not only focusing on the system development this must supplement Scrum. We learn that NHH has found it important to conduct trainings for the users so that they will both be interested in and able to use the new portal once it is made available. If this is not done they are most likely going to receive much more dissatisfaction among the users, as well as resistance toward starting to use it. Thus the PSO perspective is crucial to the success of the project.

A project is not transparent just by tracking performance and progress and putting it in a computer system. The information must be made easily available and understandable for those who request it.

In the portal project the developers keep track of all progress information on both the product backlog and the sprint backlog. They know at anytime which tasks are completed, and which are not. In addition they can see how many hours were estimated, and compare with number of hours used. They have all the information need in order to generate a burndown chart, yet if this exists, it does not appear to have been made available to everyone since respondents lack this kind of information. Other interviewees have stated that they miss information or lists that I have seen later actually do exist. This proves that even though the information exists, there has not been put enough emphasis on making it available or easy to understand. It does not matter if the information exists or not if the one looking for it either cannot find it, does not know about its existence, or does not understand it.

Agile software development and Scrum are technical project management methods which work well on coding, and thus the system development perspective of a project. The people and organization development perspective are non-technical and works well with milestones and traditional project management.

As discovered in the analysis, Scrum is mainly concerned with the development of the system, and measures progress on the code, which is in fact only about the system development. Yet the entire IT project is not only about the system, it is about people and organization development as well. Since Scrum does not take these aspects into consideration, we must rely on traditional project management to serve these areas instead, thus milestone planning as in Andersten et al.'s (2004) GDPS serves as a good starting point, and I have tried to show how we can combine milestone planning with Scrum without that affecting the way Scrum works on the system development.

Theory and practice is not the same, and although a small project, the portal project has proved interesting to study, and I believe the study has taught interesting lessons on how IT project management in practice differ from the theories. It will be exciting to read how the operation portal will be welcomed by the users when launched, as well as to see how it evolves thereafter as the Scrum sprints continue.

7. Suggestions

I would like to round off my thesis with a few suggestions to the portal project since they were so kind to make themselves available for research. But I would also like to offer some suggestions for further research, as I have found that some of my findings and thoughts could benefit from being studied even more.

First I would like to suggest that the portal project considers the transparency and availability of information in the project. This might not be an issue for all project members, but it is for some, and it can have a negative impact on motivation among project members when it difficult to get an overview of the project. One should look at the ways members of the project can get information on the status of tasks and features that are currently worked upon, as well as those in the to-do list and the ones already completed. The priorities and the product backlog should be easily available. This can for instance be solved by making this information available in the project's own SharePoint project portal.

It would be interesting if the portal project tried to implement the use of milestone on the people and organization development tasks in the project, similar to the way I have illustrated in chapter 5.6.4. I am curious if this might prove to eliminate members feeling a lack of overview of the project. I am also curious if this can be done without interfering with the Scrum processes, and if ultimately the portal project will find such a solution to be attractive and valuable.

Project members and stakeholders should also be made aware that the benefits of iterative development with Scrum and a product backlog should be better experienced in the next phase of the project, when the foundation of the portal is already completed. It is limited to what degree NHH can impact the development in this early stage of development relative to how it is expected to change as the project evolves.

Regarding further research I would suggest looking more on how one can combine Agile Software Development and/or Scrum with a PSO perspective. It could be interesting both to do more case studies on other projects (perhaps larger and in private companies) and to perform quantitative research in order to see if it can fit all kinds of IT projects.

At last I would like to wish both NHH and Bouvet good luck with the remaining work on the portal project. It will be very exciting to see the new portal when it is completed.

Bibliography

Agile Alliance (2001): “Agile Manifesto”, <http://www.agilemanifesto.org/>, Agile Alliance (26.02.2009)

Andersen, Erling et al. (2004): “Goal Directed Project Management”, 3rd Edition, Kogan Page.

Beedle, Mike et al. (1999): “SCRUM: An extension pattern language for hyperproductive software development”, Pattern Languages of Program Design, pp. 637-651, Addison-Wesley.

BNET (2009): “Definition of Leavitt's Diamond”, <http://dictionary.bnet.com/definition/leavitt%2527s+diamond.html>, BNET.com, (16.02.2009)

Bouvet ASA (2009): “Bouvet Web Site”, <http://www.bouvet.no/>, Bouvet ASA, (30.03.2009)

Brooks, Frederick P. (1995): “The Mythical Man Month”, 2nd Edition, Addison-Wesley.

Cockburn, Alistair (2002a): “Learning From Agile Software Development - Part One”, October 2002, pp 10-14, Crosstalk: Journal of Defense Software Engineering.

Cockburn, Alistair (2002b): “Learning From Agile Software Development - Part Two”, November 2002, pp 9-12, Crosstalk: Journal of Defense Software Engineering.

CPA (2009): “Glossary of Poverty and Social Inclusion Terms”, <http://www.cpa.ie/povertyinireland/glossary.htm>, CPA.ie, (23.04.2009)

Dahl, Audun A. and Anna M. Fuglseth (2007): “Informasjonssystemer og økonomisk styring”, BUS431 Lectures, Fall 2007, NHH.

Danilova, Kjersti B. and Arkadi Danilov (2008): “FS' rolle i New Public Management”, NHH.

Dinkla, Jörn (2009): “Pragmatic Programming”, <http://www.dinkla.net/development/pragmatic.html>, Dinkla.net, (25.03.2009)

Gilb, Tom (1988): “Principles of Software Engineering Management”, 1st Edition, Addison-Wesley.

Gilb, Tom (1976) “Software Metrics”, Little, Brown, and Co.

Grude, Kristoffer V. (2007): “Goal Directed Project Management as a practical method”, Guest lecture, VOA026, NHH, 16.-17.03.2007.

Highsmith, Jim (2002): “What is Agile Software Development?”, October 2002, pp 4-9, Crosstalk: Journal of Defense Software Engineering.

Highsmith, Jim and Alistair Cockburn (2001a): “Agile Software Development: The Business of Innovation”, Vol. 34, No. 9, pp 120-122, September 2001, IEEE Computer, USC.

Highsmith, Jim and Alistair Cockburn (2001b): “Agile Software Development: The People Factor”, Vol. 34, No. 11, pp 131-133, November 2001, IEEE Computer, USC.

Holmesland, Tore B. (2007): “IT Project Management”, VOA026 Lectures, Spring 2007, NHH.

IT User Forum (2007/01): “IT-brukerforum 2007/01”, <http://www.nhh.no/no/om-nhh/avdelinger/it-avdelingen/it-brukerforum/2007/referat-fra-møte-i-it-brukerforum-2007/01.aspx>, NHH.no, (05.02.2009)

Larman, Craig and Victor R. Basili (2003): “Iterative and Incremental Development: A Brief History”, Vol. 36, No. 6, pp 47-56, June 2003, IEEE Computer, USC.

Leavitt, Harold J. (1965): “Applied Organizational Change in Industry: structural, technological, and humanistic approaches”, Handbook of Organizations, J.G. March, Rand McNally, Chicago.

Marthinsen, Kristian T. (2008a): “Stor digital satsing for NHH”, <http://paraplyen.nhh.no/paraplyen/arkiv/2008/februar/stor_digit/>, Paraplyen, (14.02.2008)

Marthinsen, Kristian T. (2008b): “Hva skjer med portalprosjektet?”, <http://paraplyen.nhh.no/paraplyen/arkiv/2008/oktober/hva_skjer_/>, Paraplyen, (17.10.2008)

Marthinsen, Kristian T., (2009): “Norges beste virksomhetsportal?”, <http://paraplyen.nhh.no/paraplyen/arkiv/2009/februar/norges_bes/>, Paraplyen, (09.02.2009)

Merriam, S.B. (1988): “Case-Study Research in Education”, Jossey-Bass Inc. Publishers.

NHH Web Site (2009): “IT Department”, <http://www.nhh.no/en/about-nhh/administration/it-department.aspx>, NHH.no, (05.02.2009)

Paraplyen (2008): “Stor interesse for portalprosjektet”, <http://paraplyen.nhh.no/paraplyen/arkiv/2008/mars/stor_inter/>, Paraplyen, (05.03.2008)

Repstad, Pål (1991): “Mellom nærhet og distanse: Kvalitative metoder i samfunnsfag”, 2nd Edition, Universitetsforlaget.

Royce, Winston W. (1970): “Managing the Development of Large Software Systems”, Vol. 26, pp 1-9, August 1970, Proceedings, IEEE WESCON.

Schwaber, Ken (1995): “SCRUM Development Process”, 6(4), October 1995, ACM/SIGPLAN OOPS Messenger.

Schwaber, Ken (2007): “What Is Scrum?”, <http://www.scrumalliance.org/resources/227>, Scrum Alliance, (27.03.2009)

Shojaee, Hamid (2008): “SCRUM in Under 10 Minutes”, <http://www.youtube.com/watch?v=Q5k7a9YEoUI>, Axosoft, YouTube, (09.12.2008)

Sutherland, Jeff (2004): “Agile Development: Lessons Learned From the First Scrum”, Vol.5, No. 20, Executive Update, Cutter Agile Project Management Advisory Service.

Turk, Dan et al. (2002): “Limitations of Agile Software Processes”, XP2002, pp 43-46, Springer-Verlag.

WikiPedia (2009a): “Agile Software Development”, <http://en.wikipedia.org/wiki/Agile_software_development> (and sublinks), Wikimedia, (06.01.2009)

WikiPedia (2009b): “Semi-structured interview”, <http://en.wikipedia.org/wiki/Semi-structured_interview>, Wikimedia, (24.04.2009)

WikiPedia (2009c): “SCRUM”, <http://en.wikipedia.org/wiki/SCRUM>, Wikimedia, (26.03.2009)

Winter, Mark et al. (2006): “Focusing on business projects as an area for future research: An explanatory discussion of four different perspectives”, Vol. 24, pp 699-709, International Journal of Project Management.

Appendices

Appendix I: Interview with Project Manager at NHH

This interview took place at NHH in Bergen on Monday, March 9th 2009.

This was the first interview that was conducted, and thus there were two main purposes: the first being to get an overview of the project and learn as much as possible about what it was all about, and the second being to learn about how the project was being managed from the Project Manager's perspective.

Overview of the project:

- What is the purpose of the project? What does it include and involve?
- What is the background for the project?
- How does the project time line look like?
- Who are the project members, what are their roles and relationships?
- How is the Operational Portal defined in the Portal Project?
- Generally let the interview object talk freely about the project.

Project Management:

- Which project management model is being used? And how?
- How is the project structured and organized?
- How do all of you work together in this project?
- About planning before and during the project. About the public tender.
- Costs. Resources. Budget.
- How is the experience with Agile development methodology (Scrum) so far?
- Positive experiences working with this project?
- What challenges are experienced?
- Reporting? Meeting? Documentation? Control? Progress?
- Comparing with previous project experience and development.

Appendix II: Interview with Technical Sub-Project Manager at NHH

This interview took place at NHH in Bergen on Monday, March 9th 2009. It was not prepared in advance, but took place as the interview object offered to do the interview on short notice, i.e. just after the interview with the project manager.

The most interesting about interviewing the technical sub-project manager, is that I would get information directly from the person who has the daily meetings with the developers. This would allow me to hear about how this person experiences Scrum, being directly involved in scrum processes daily yet having no former experience with Scrum.

Some of the topics discussed in the interview:

- About the role of the Technical Sub-project (compared with the two others).
- Former experience with technological development.
- Experience with Scrum and working with Bouvet.
- Positive and negative experiences with Scrum.
- About the daily Scrum meetings with developers.
- Generally about meetings in the project.
- Why Scrum?
- Generally let the interview object talk freely about the project.

Appendix III: Interview with Content Sub-Project Manager at NHH

This interview took place at NHH in Bergen on Thursday, March 13th 2009.

The Content Sub-Project Manager has main responsibility for the information and the content, and the way it will be displayed on the portal. As opposed to the other sub-project managers, the Content Sub-Project Manager is employed in NHH's information department, and not in the IT department.

Some of the topics discussed in the interview:

- About the role of the Content Sub-project (compared with the two others).
- How is work and tasks conducted in the content part of the project?
- In what ways is the content part affected by the Scrum methodology?
- How is information going to be easier accessible in the new portal?
- What makes the portal better with regards to relevant content and information?
- What have been done in order to inform researchers, employees, and students about the portal project?
- What have been done in order for researchers, employees, and students to influence the new portal?
- How do you focus on the people and the organization in addition to the development of the system?
- To what degree has the content part influenced the project so far?
- How do you plan to conduct trainings on using the new portal when released?
- Positive effects and challenges working with Scrum.
- Generally let the interview object talk freely about the project.

Appendix IV: Interview with Developers/Programmers at Bouvet ASA

This interview took place at Bouvet ASA in Stavanger on Friday, March 20th 2009.

The interview was mainly conducted with one of the two developers. However, due to both of them being located in the same room, the second programmer contributed occasionally as well. Some parts of the conversation included showing examples on the computer screens serving as valuable illustrations, and as good compensation for technical language.

Some of the topics discussed in the interview:

- How do you program and develop this project on a daily basis?
- How do both of you work together? Sharing tasks and responsibility?
- Why do you use Scrum? And how do you use Scrum? Explain Scrum Light?
- How do you keep track of the status of the project backlog and sprint backlog?
- How do you report on progress and keep NHH informed?
- Compared with similar projects worked on earlier, how do you expect the project to change?
- Public vs. private customers. NHH as a customer.
- How is the portal project different from other IT projects?
- Generally let the interview object talk freely about the project, especially on the technical part of the project, and how working methods are arranged to practically fit the Scrum method.

Appendix V: Interview with a member of Project Support Team

This interview took place at NHH in Bergen on Friday, March 13th 2009.

The interview object was a scholarship holder at the NHH's Strategy and Management department, and was of particular interest due to former experience from managing IT projects in past employment.

Some of the topics discussed in the interview:

- About the interview object's role and the project support team's role.
- What thoughts do you have on Scrum and using Scrum on the portal project?
- From former project experience, what are your views on Scrum? And how does Scrum fit with the portal project?
- How is the portal project different from other projects you have worked on?
- In what ways do you keep a PSO perspective in the project?
- Public vs. private customers. NHH as a customer.
- Generally let the interview object talk freely about the project.
- Generally let the interview object talk freely about previous IT project management experiences, and lessons learned from this, both positive and negative.