The effects of Kanban in software development teams-
a study of the implementation at Sandvik

Effekterna av Kanban i systemutvecklingsteam -
en studie av implementationen på Sandvik

Robin Ericsson
Anna Granlöf

2011-06-16
EXAMENSARBETE,
Grundnivå 2 i Informatik

<table>
<thead>
<tr>
<th>Ämne</th>
<th>Reg nr</th>
<th>Omfattning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informatik, Grundnivå 2</td>
<td>IKA042011</td>
<td>15 hp</td>
</tr>
</tbody>
</table>

Namn
Robin Ericsson
Anna Granlöf

Månad/Ar
Juni 2011

Handledare: Pär Douhan & Amra Halilovic
Examinator: Bo Sundgren

Företag/Institution
Sandvik/Sogeti

Handledare vid företaget/institutionen
Christophe Achouiantz & Björn Thune

Titel
Effekterna av Kanban i systemutvecklingsteam -
en studie av implementationen på Sandvik

Nyckelord
Kanban, Sandvik, effekter, systemutveckling

Sammanfattning
# DEGREE PROJECT,
**Undergraduate level 2 in Informatics**

<table>
<thead>
<tr>
<th>Subject</th>
<th>Reg number</th>
<th>Extent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informatics, Undergraduate level 2</td>
<td>IKA042011</td>
<td>15 ects</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Names</th>
<th>Month/Year</th>
<th>Supervisor at the Company/Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robin Ericsson</td>
<td>June 2011</td>
<td>Christophe Achouiantz &amp; Björn Thune</td>
</tr>
<tr>
<td>Anna Granlöf</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Company/Department</th>
<th>Supervisor at the Company/Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sandvik/ Sogeti</td>
<td></td>
</tr>
</tbody>
</table>

| Title | |
|-------| |
| The effects of Kanban in software development teams - a study of the implementation of at Sandvik | |

| Keywords | |
|----------| |
| Kanban, Sandvik, effects, software development | |

## Summary

In software development organizations there is sometimes a need for change. In order to meet continuously increasing demands from their customers, Sandvik IT Services - SITS, at Sandvik in Sweden, required improving the way they worked with software development. Due to issues like a lot of work in progress and lot of simultaneous tasks for individuals in the teams that caused stress, it was almost impossible to address the question of working with improvements. In order to enable the improvement process Kanban was introduced in the software development teams. Kanban for software development is a change method created by David J. Anderson. The purpose of this thesis is twofold. One part is to assess what effects Kanban has had on the software development teams. The other part is to make a documentation of the Kanban implementation process at SITS. The documentation has been made on the basis of both company internal resources and observations of the Kanban implementation process. The effects of Kanban have been researched with an interview survey to the teams that have gone through the Kick start of the Kanban process. The result of the thesis is also twofold. One part of the result is an extensive documentation of the implementation process of Kanban at SITS. The other part is an assessment of the effects that Kanban has had at SITS. The major effects have been that the teams are experiencing less stress, more focus on quality and better customer collaboration. It is also evident is that it takes time for some effects to evolve when implementing Kanban.
Preface

This thesis is written in the course IK2009 Thesis for Bachelor of Informatics degree at Dalarna University in Borlänge, Sweden.

We would like to express our gratitude to several persons that have made this study possible. First and foremost we would like to give our heartfelt thanks our instructors at the university Pär Douhan and Amra Halilovic, and our instructors at Sogeti Christophe Achouiantz and Björn Thune. You have all given us very valuable advice and the support necessary for us to be able to carry this work through. We would also like to give Johan Nordin at Sandvik a lot of thanks for contributing with both his deep knowledge and good ideas. Our warmest thanks go to all the employees at Sandvik for being very generous in sharing their experiences and thoughts with us. We would also like to thank all the other persons that have helped us and contributed to our report like our families and our opponents.

Borlänge June 16th 2011
## Contents

1 Introduction .......................................................................................................................... 1  
  1.1 Background .................................................................................................................... 1  
    1.1.1 Assigners .................................................................................................................. 2  
  1.2 Problem formulation ....................................................................................................... 2  
  1.3 Purpose .......................................................................................................................... 2  
  1.4 Objectives ..................................................................................................................... 3  
  1.5 Delimitations .................................................................................................................. 3  
  1.6 Method overview .......................................................................................................... 3  
  1.7 The outline of the report ............................................................................................... 3  

2 Methodology ......................................................................................................................... 4  
  2.1 Method theory ................................................................................................................ 4  
    2.1.1 Hermeneutics and qualitative methods ................................................................. 4  
    2.1.2 The theoretical and the empirical relationship ..................................................... 4  
  2.2 Choice of methods .......................................................................................................... 4  
    2.2.1 Literature studies .................................................................................................... 5  
    2.2.2 Observations of the work processes ..................................................................... 6  
    2.2.3 Survey interviews .................................................................................................. 6  
    2.2.4 Analyzing methodology ....................................................................................... 7  
    2.2.5 The display- and analysis model ......................................................................... 8  

3 The development of Agile, Lean and Kanban ..................................................................... 9  
  3.1 The historic development of Agile methods ................................................................ 9  
    3.1.1 The Waterfall model ............................................................................................... 9  
    3.1.2 The Spiral model and Rational Unified Process ................................................. 10  
    3.1.3 The Agile models and the Agile Manifesto ......................................................... 11  
  3.2 Lean ............................................................................................................................... 12  
    3.2.1 Lean history .......................................................................................................... 12  
    3.2.2 Lean thinking in software development ............................................................... 13  
  3.3 Kanban .......................................................................................................................... 14  
    3.3.1 The emergence of the Kanban method ................................................................. 14  
    3.3.2 The five core properties of a Kanban implementation ........................................ 15  
    3.3.3 Implementing Kanban ......................................................................................... 15  
    3.3.4 Effects of Kanban ................................................................................................. 16  
  3.4 Why choose Kanban? ...................................................................................................... 17  
  3.5 The connection between Software development methods, Kanban and Lean ............... 17  
  3.6 Our view of how Software development methods, Kanban and Lean are connected .... 18  

4 Why Kanban at Sandvik? ................................................................................................... 19  
  4.1 The Systems development Office – SDO ................................................................... 19  
  4.2 The concept of a common software development process ......................................... 19  
    4.2.1 Solution design ...................................................................................................... 20  
    4.2.2 Develop feature ................................................................................................. 20  
    4.2.3 Release to production ......................................................................................... 21  
  4.3 The concept of Definition of Done ............................................................................... 21  
  4.4 Workshops .................................................................................................................... 21  
  4.5 Problems in software development teams ..................................................................... 22  
  4.6 Reasons for introducing Kanban .................................................................................. 23
Table- and figure contents

Table 1. Empirical evaluation model ................................................................. 8
Table 2. The effects on task switching ............................................................... 37
Table 3. The effects on technical debt, quality and incidents, and rework .......... 38
Table 4. The effects on customer organization ............................................... 39
Table 5. The effects on the time aspect ............................................................ 40
Table 6. The effects on satisfaction ................................................................. 42
Table 7. Summary of the effects of Kanban at Sandvik ................................... 43
Table 8. The effects reached with Kanban within two months ...................... 45
Table 9. The effects reached with Kanban between three and five months ...... 46
Table 10. The effects reached with Kanban after six months ....................... 47

Figure 1. The outline of the report ................................................................. 3
Figure 2. Visualization of the work process of the report ............................... 5
Figure 3. The Waterfall model .................................................................... 9
Figure 4. The Spiral Model ......................................................................... 10
Figure 5. Our graph of the connections between software development methods, Kanban and Lean ............................................................... 18
Figure 6. SITS systems development process ................................................ 20
Figure 7. Problems in software development at Sandvik in 2010 ................. 23
Figure 8. The SDO improvement concept ...................................................... 24
Figure 9. Sandvik’s roadmap towards Kanban .............................................. 26
Figure 10. A team’s policy flip chart ............................................................. 28
Figure 11. Description of a team’s work item ............................................... 29
Figure 12. Standard SITS Kanban board ....................................................... 30
Figure 13. A Kanban board with the definitions of done at the bottom ........ 31
Figure 14. Visualizing priorities with swim lanes and using avatars ............. 33
Figure 15. The Retrospective board ............................................................. 35
Figure 16. Our summary of Kanban at Sandvik ........................................... 35
1 Introduction
Here we will start by giving a background to our report and our assigners then we will describe the problem, our purpose and objectives, a method overview and the outline of the report.

1.1 Background
In everyday life today we use a lot of things that contains software both at home and in the industry. Processes and development methods are used to develop the software in order to make the development processes as efficient as possible. These methods will sometimes need to be changed when the outcome of the used methods are insufficient. For a more extensive explanation of Agile software development methods and Lean software development see 3.1 and 3.2.

“The pertinent mission of software project management is to continuously achieve more and more successful projects. In the field of software development, the Kanban method has gained momentum recently, mostly due to its linkages to Lean Thinking.” (Ikonen, Pirinen, Fagerholm, Kettunen & Abrahamsson, 2011).

Christophe Achouiantz at the Systems Development Office at Sandvik says “Introducing an Agile or Lean system and/or software development method in an organization formed after a traditional development model is difficult. The risk being that the change does not stick and the organization ultimately rejects the new way of doing development, even after several successful project pilots.” (personal communication, April 20, 2011).

Kanban for software development, as described by David J. Anderson (2010a) in "Kanban - Successful Evolutionary Change for Your Technology Business", is designed to be evolutionary and adaptable in the changes it brings. This approach gives organizations the opportunity to adapt and implement the method to fit their own needs and at the rate that the teams can cope with.

Using the Kanban change method requires thinking in terms of visualization of the work flow, well defined policies, establishing flow, limiting work in progress and improvements. Teams using traditional software development methods are required to start thinking differently. Implementing Kanban fully - and getting its full effect - requires the teams to mature in agility and Lean thinking. (Anderson, 2010a).

In order to meet continuously increasing demands from their customers, Sandvik IT Services, SITS, required improving the way they worked with software development. The Systems Development Office’s, SDO’s, mission was to improve and standardize the way of working at SITS. For more information on the SDO see 4.1.

During the pre-improvement process a general observation was made, some issues were regularly occurring; there were a lot of things going on at the same time, when things got stuck new things were started causing a lot of work in progress and a lot of simultaneous tasks for individuals in the team. The consequences of that were mainly that it caused a lot of stress and made it nearly impossible to address the question of working with improvements. In order to enable the improvement process, Kanban was introduced as a complementary tool to help teams see, understand and later control the work in progress, with the end goal to enable improvement activities. SITS is now interested in an evaluation of the effects that the Kanban introduction has had.
1.1.1 Assigners
The work regarding the implementation of Kanban is operated by SITS with the help of a consultant from Sogeti. This report is sponsored by Sogeti and the work is executed at SITS at Sandvik in Sandviken, Sweden.

Sandvik AB is a Swedish engineering company. The company is divided into three main areas: Tooling, Materials Technology and Mining and Construction. Sandvik AB has about 47 000 employees worldwide, the main office is located in Sandviken, Sweden, where the company was founded in 1862. The company is world leading in tools for metal cutting, equipment and tools for the mining and construction industries and products in advanced stainless materials, titanium, special alloys, metallic and ceramic resistance materials and process systems.

Sandvik IT Services (SITS) is a part of Sandvik AB. SITS has approximately 900 employees worldwide and 500 of them are located in Sweden. SITS mission is to provide support, increase efficiency and keep a problem free environment for Sandvik AB and their IT-users. (Sandvik, 2011).

Sogeti is an international company that specializes in IT consulting. The company is located in 15 countries and has about 20 000 employees worldwide. The company is present at 21 Swedish locations and has about a 1000 employees in Sweden. Some of the services the company provides are software development, project management, testing and system administration. (Sogeti, 2011).

1.2 Problem formulation
Sandvik IT Services decided to use Kanban as a tool to create a sustainable pace in the development teams. Sustainable pace is important in order to initiate the work with raising maturity in the software development teams. The effects and trends seen so far indicates that there are more positive effects coming from the implementations in the software development teams. The Systems Development Office, SDO, is now interested in an assessment of the effects of the Kanban implementations so far.

The introduction of Kanban has been rapid at Sandvik IT Services. The SDO’s work has been concentrated on Kick starting the teams that have requested to use Kanban and supporting the teams already in the process of using Kanban. The SDO started helping the first team in using Kanban during the fall of 2010. As of May 2011, the SDO has Kick-started 20 teams. For more information on the Kick start, see 5.2. Based on the positive response of these teams, another 10 teams are requesting to get support in using Kanban. The SDO is now interested in evaluating the effects of Kanban for these teams as well as the effect of the current way of coaching Kanban. Based on these effects, the SDO is ready to adapt its coaching strategy for better effects.

Being busy Kick starting and supporting teams, the SDO has not had enough time to make an extensive documentation of its own work. A need for a more extensive documentation of the implementation process has grown stronger and stronger. The documentation is important for both SITS and the SDO in order to be able to gather the current knowledge of Kanban and to make knowledge transferring easier.

1.3 Purpose
The purpose of this study is to make a contribution to the understanding of the implementation process of Kanban in software development teams and the effects that Kanban can have in software development teams.
1.4 Objectives
The objectives are:
- to assess the effects of Kanban in the software development teams at SITS
- to document the implementation process of Kanban in software development teams at SITS

1.5 Delimitations
We limit the study to be executed at SITS and their approach on introducing Kanban to the software development teams. The Kanban methodology assessed is the one created by David J. Anderson (2010a) for software development. The results will be generated from SITS approach to Kanban and might not have general applicability to other companies.

1.6 Method overview
Literature studies will extend throughout the whole work process. Data gathering will be done through literature studies, observations and interviews. We will conduct survey interviews at SITS with members of teams in the process of implementing Kanban. The data will be analyzed and assessed against the theory, and then put in our analysis- and display model. We will present the report for students at Dalarna University and for the staff at both SITS and Sogeti.

1.7 The outline of the report
This report will be outlined as seen in Figure 1. The report will be divided into four main parts.

![Figure 1. The outline of the report (freely after Björklund & Paulsson, 2003)](image)

First is the introductory part that includes the introduction and the methodology. Then there is the theoretical part that includes the development of Agile methods, Lean software development and the Kanban methodology. The empirical part includes why Kanban is used at Sandvik, Kanban at Sandvik and the effects of Kanban at Sandvik. From the theoretical and the empirical part the analysis is done which is a comparison of the theoretical effects and the actual effects of Kanban at Sandvik. The last part of the report is our conclusions.
2 Methodology

This chapter is about the methods that we used to gather the information about the Kanban implementation process and then how we studied and analyzed it. The following part motivates the choices of methodology for this report.

2.1 Method theory

In developing knowledge the process is based on our basic understandings of science. Different methods make different assumptions about the object of the study. Different methods also make different takes on the problem. According to Patel and Davidson (2003) today there are two main scientific approaches; positivistic and hermeneutical.

2.1.1 Hermeneutics and qualitative methods

We have chosen to use the hermeneutical approach, also expressed as the science of interpretation, because we are seeking a comprehensive understanding of what we are studying. We have tried to interpret and understand Kanban both through language, both written and spoken, and through actions, by observing the Kanban implementation process. The hermeneutical approach is closely connected to qualitative methods. In this report we have used qualitative methods since they focus on the exploration of “soft” data. They also often use verbal analysis methods of written texts, which we have done by our survey interviews. We have tried to interpret and understand the concept of Kanban. This is what Patel and Davidson (2003) refers to as a hermeneutical approach and a qualitative method.

2.1.2 The theoretical and the empirical relationship

The data and information about the reality that is gained through studies are often referred to as empirical. The researcher’s role is to relate theory and reality to one another. There are three main ways to do that and they are by using deductive, abductive and inductive reasoning as explained by Patel and Davidson (2003).

We chose an inductive way because we could study Kanban at Sandvik without anchoring it in an established theory and follow our path of discovery. Even though we have not based our studies on an established theory our own ideas and notions have influenced our work and we have based our results on our empirical studies of a specific case, the Kanban implementation process at Sandvik. When working inductively the result of the research will suggest the “truth” but not guarantee it and the result might not have general applicability, just as Patel and Davidson (2003) explain.

2.2 Choice of methods

In order to do our research the best way we see possible, we have chosen to use mainly a hermeneutic approach. We have also chosen to use qualitative methods to collect and then study and analyze the data with a phenomenographic approach, 2.2.4.

The validity of the research can be increased by using triangulation, i.e. using several different methods to examine the same topic – method triangulation, and several different data sources – data triangulation (Björklund & Paulsson, 2003). We have used method triangulation since we have studied the topic using literature studies, observations and interviews. We have also used data triangulation by using different sources to collect the data; the observations and of course the interviewees have been an important source of both knowledge and data.
Figure 2. Visualization of the work process of the report (freely after Björklund & Paulsson, 2003)

The practical work procedure has been divided into the phases displayed in Figure 2. We started by beginning to understand our mission then used unstructured interviews with our assigners to deeper understand and establish our mission. During the whole work process we have been conducting literature studies. We wrote the theoretical part and after that made observations and survey interviews in order to be able to write the empirical part which we then used for our analysis.

2.2.1 Literature studies

Literature studies were the first step in creating a basic knowledge of our topics. The studies of literature have been present during our entire work process. We have assessed and evaluated the empirical data against our theoretical knowledge and understandings throughout the work process. We have repeatedly reviewed our sources in order to try to create a more knowledgeable foundation for interpreting our empirical data in an iterative process.

When studying an IT interrelated topic the Internet is an excellent source of recent articles, reports, literature and thoughts. The topics of our report are no exception there are several blogs and informal communities that discuss Lean, Agile and Kanban methodologies. Since several of the forerunners teach and coach the techniques for a living, they might be favoring their own methodologies and views.

It is also pointed out by Björklund and Paulsson (2003) that literature is secondary data, the information in the literature is gathered and presented with a purpose that might not be the same as the studied topic. That is why it is necessary to question both the information and the way the material is used.

We have to the greatest extent possible used the criteria for source analysis as explained by Leth and Thurén (2000) as guidance for our assessment of sources. The four criteria are time, dependence, authenticity and bias. Concerning the criteria of time they say that it is most important to check when the website last was updated to make sure that it is up to date. It is also important to check what dependencies can be found concerning both to other sources and handovers. They point out that it is important to if possible go back to the primary source. The authenticity criteria mean that it is important to check that the source is actually what it claims to be. Sources can be biased every time someone has an interest in something the bias can be both concerning facts and explanations.

Literature studies were important in our study to give us the necessary both general and specific theories for a solid background. The literature studies also provided us with knowledge on how to analyze and evaluate the topic.
We chose to use literature studies since it gave us the opportunity to quickly study a lot of information about the Lean, Agile and Kanban methods and to give us a relatively easily accessible source of information for our theoretical frame of reference. The results of our literature investigations regarding system development methods, Lean and Kanban can be seen in chapter 3.

2.2.2 Observations of the work processes
To add to the knowledge from the literature studies we also chose to make observations. Björklund and Paulsson (2003) describe two ways of observing an activity. One way is watch it as an outsider without taking part in the activity. The other way is to be an active participant in the process, to make participation observations. We have used both kinds of observations in our study.

The kinds of observation we have made are called unstructured observations (Patel & Davidson, 2003). These observations are used in an exploratory way to gather as much information as possible. When we were in the observation situations we as observers wrote down key words individually. At the end of the observation day we put together a longer and more extensive summary of our combined observations.

These observations have been essential for us to gain the personal understanding of how the daily work is carried out in the work teams. We made several observations of Kanban boards, stand-up-meetings, and retrospectives and also observed three teams go through their Kick starts. In these observation sessions we got the opportunity to study the processes and attitudes in the teams. We were then able to assess and validate these perceptions in the interviews.

We chose to use observations to give us an opportunity to make a firsthand study of the work processes in a more objective way and the observations were invaluable primary sources. At the same time the drawbacks of these observations were that they were time consuming, this drawback is also expressed by Björklund and Paulsson (2003) but we felt that the advantages of observations outweighed the disadvantages.

2.2.3 Survey interviews
The main method we chose for gaining qualitative empirical data was a combined interview and survey. We wanted to get deeper knowledge and understanding by getting the thoughts directly from the team members. As pointed out by Patel and Davidson (2003) interviews are often conducted face-to-face but can also be varied to fit special needs.

The survey interviews were performed individually and in a semi-structured way, where the questions were prepared in advance, see appendix 1, and accompanied with supplementary questions (Björklund & Paulsson, 2003). The questions were open ended questions. The survey interviews were performed at SITS. The survey interviews were sent to all the software development teams that have gone through the Kanban Kick start. The survey interviews were inserted into the survey tool at SITS Intranet and distributed electronically to the 12 teams. The questions of the survey interviews were accompanied by a covering letter explaining and motivating the value of the interviewee’s participation.

The drawbacks of not having the interviewees face-to-face when conducting the survey interviews are that we were not able to see and interpret the body language of the interviewees. We could not either adapt each question to each individual respondent or individually adapt the supplementary questions as expressed by Björklund and Paulsson (2003). Another drawback can be that some of the answers might have been misinterpreted by us.
Although there were drawbacks the advantages in our opinion far outweighed the drawbacks, that is why we chose to use survey interviews since the interviewees, as expressed by Björklund and Paulsson (2003) supplied us with first hand information that was pertinent to our study, the interviewees were our most important primary sources.

The survey interviews allowed us in a shorter period of time to collect more data than if we had done the interviews face-to-face. The survey interviews were conducted electronically and the interviewees responded in writing. The survey interviews were anonymous (Patel & Davidson, 2003) since we could not identify the individual respondent. The interviewees had one week to supply their answers to the survey interviews. After collecting the answers we then analyzed the answers.

2.2.4 Analyzing methodology
We first got acquainted with the written answers to the qualitative survey interviews to get an overall impression. Then we observed what similarities and differences we could find in the answers. After that we categorized the perceptions into description categories and finally we tried find the underlying structure of the description categories. This process is what Patel and Davidson (2003) describes as the phenomenographic analysis.

In this process we used an inductive approach where the material is studied and sorted until patterns are visible. The pattern was divided into categories in which the statements from the interviews were sorted. This pattern finding process was iterative and consisted of a continual sorting and resorting of the data. The developing categories of description is supposed to be so clear cut that it is definite to which category a statement belongs. We later found out that this was not as easy in practice as it sounds in theory. Since perceptions regarding a phenomenon, in our case Kanban, can be changed through learning, growth and progress the result of a phenomenographic approach is considered a local theory since our result is generated from a single study of Kanban at Sandvik. This way of working is supported by Patel and Davidson (2003).

We will use both inductive and abductive reasoning while we combine the methodical analyzing from the phenomenographic approach with the hermeneutic approach where we use our preconceived notions to make a deeper and more comprehensive analysis because we feel that this approach is the best suited for our research.
### 2.2.5 The display- and analysis model

In order to display the results of the analysis we use the table in Table 1 to visualize the results.

<table>
<thead>
<tr>
<th>Problematic topics before the Kanban implementation</th>
<th>Evaluation of these topics after the Kanban implementation</th>
<th>Expected influences of Kanban</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXX</td>
<td>XXX</td>
<td>XXX</td>
<td>Supported</td>
</tr>
<tr>
<td>YYY</td>
<td>YyY</td>
<td>Yyy</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>ZZZ</td>
<td>ZZZ</td>
<td>OOO</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

Table 1. Empirical evaluation model (freely after Ikonen et al., 2011, p.8)

The table is used to display the problematic topics the work teams had before the implementation of Kanban, the evaluation of these topics after the implementation of Kanban, the expected influences of Kanban according to the theory, see 3.3.4 Kanban and to what degree the outcome is supported.
3 The development of Agile, Lean and Kanban

Here we will give a theoretical background and presentation of related work to provide an improved understanding of our topic. We will start by presenting the development of Agile methods, then the Lean software development and the Kanban methodology. This will give a background on how to go from Agile or traditional system development methods to Lean using Kanban.

3.1 The historic development of Agile methods

This part of the theory chapter will describe some of the different methods in software development leading up to Agile methods as well as describing the Agile methods and the Agile Manifesto.

3.1.1 The Waterfall model

In the 1970’s the software development process the Waterfall model was presented by Royce (1970). The Waterfall model is a process of software development where progress flows, like a water fall, downwards through the phases in sequential order; system requirements, software requirements, analysis, program design, coding, testing and operations as seen in Figure 3.

“I believe in this concept, but the implementation described above is risky and invites failure” are Royce’s own words about his reservations of the Waterfall model. (Royce, 1970, p.2)

Ladas (2008) describes how Royce wanted to create a more feedback-driven model, but the sequential Waterfall model was very appealing to the business culture in the 1970’s since it was a very easily controlled process for the management.
The model was fully diffused in the business culture and stuck without concern about the initial reservations that Royce himself had. After a relatively long time reactions were voiced, one of the most forceful belonged to Barry Boehm.

3.1.2 The Spiral model and Rational Unified Process
Boehm (1986) in some ways went back to the initial thoughts of Royce about the necessity of having overlapping phases instead of sequential phases in the software development process. Boehm wanted the process to be cycle driven, as seen in Figure 4, and said that the Spiral model would be more adaptable to different kinds of development projects than the Waterfall model.

![Figure 4. The Spiral Model (Boehm, 1986, p. 4)](image)

Ladas (2008) continues to describe how the concepts of using iterative development in the Spiral model then resulted in the Rational Unified Process.

Rational Unified Process – RUP promotes the use of iterative development. The development is organized into four phases; inception, elaboration, construction and transition. Each of the phases the software development process goes through consists of one or more iterations. (IBM, 2011).

According to Ladas (2008) the most important effect that RUP had was to introduce the concept of limiting the amount of iterations in software development.
3.1.3 The Agile models and the Agile Manifesto

None of the existing models were sufficient enough for the needs so software developers started working on more lightweight software development models that were based on flexibility. Focus was also put on what customers wanted and needed. (Ladas, 2008).

As early as in the 1970’s Vic Basili and Joe Turner (1975) recommended the iterative and incremental software development that is now used in Agile development methods. They wanted the developer to break the project into iterations and in that way deliver working parts of the software during the whole development process. By using an incremental development the developer was able to learn from mistakes in previous versions.

According to Fowler (2000) the change that the Agile methods brought were the compromise between no software development methods and the heavyweight methods which had too much focus on following the methodology that in turn caused the development processes slow down too much. This compromise between in some cases no process and too much process was needed and sought for by many software developers. They wanted just enough process to get the work done. Software developers needed processes that switched the focus from process orientation to people orientation. The skills of the people in the software development teams were the most important factor and the process should be a support in doing their work not an obstacle. Another disadvantage in existing methods was the inability to adapt the methods to changes whereas with Agile methods the ability to change is built in.

Gustavsson (2010) means that many of the new methods and models evolved from situations of crisis for example when project were unable to keep their deadline or the quality of the products were too low. To save the projects new methods were needed. These new methods kept a tighter control over the quality but at the same time kept the flexibility, in other words they used continuous follow-ups while still maintaining the possibility to quickly change the project plan. These new methods got the collective name, Agile methods, because they were alert, responsive and flexible.

In 2001 a group of 17 software developers and programmers gathered to discuss these Agile lightweight software development methods. Together they wrote the Agile Manifesto that lays the foundation for all Agile methods. (Agilemanifesto, 2001a):

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

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

That is, while there is value in the items on the right, we value the items on the left more.”

(Agilemanifesto, 2001b)

Larsson (2007) gives examples of some of the Agile methods: XP, DSDM, Scrum and Feature Driven Development.

Ladas (2008) says that the Agile movement has given software development a lot of progress and benefits. At the same time he says that it also has brought some big distractions. He draws the conclusion that these distractions can be substituted with something better like influences from other directions for example Lean.
3.2 Lean
In this part we describe the evolution of Lean, beginning with the Lean manufacturing model developed at Toyota and ending with the Lean software development.

3.2.1 Lean history
According to Jones, Roos and Womack (1990) the history of Lean manufacturing started in the Toyota manufacturing industry. Toyota manufactured trucks for the Japanese military during World War II. When the war ended in 1945 instead of making war material they wanted to start manufacturing cars and trucks for the commercial market.

Representatives from Toyota visited some of the big auto manufacturers in America to get inspiration on how to improve the manufacturing techniques. The Japanese were struck with how much waste of both material and time that was going on in the American factories, for example the dies that were used to shape the different metal parts of the car could just be used to a certain part of the car and it took a long time to change between different dies (Jones, Roos & Womack, 1990).

According to Jones, Roos and Womack (1990) the Toyota representatives went back to Japan with a lot of new ideas. With the new ideas and inspiration the Toyota Company started to go through some drastic changes. They started a Toyota community were employees and their families could live and have access to different recreational facilities. All employees had a chance to a lifetime employment this made the workers relaxed and kept them from leaving the company. Toyota also started an employee suggestion program; they realized that the ones that were best suited to improve the manufacturing process were the workers working in the manufacturing process every day.

In the western factories the key aspect was to keep the production line going, the workers and the foremen knew that the key factor was to produce the projected numbers of cars per day. They knew that any small glitches or defective parts were going to be fixed in the rework area at the end of the line. Instead of ignoring problems just to keep the production line going, at Toyota they would stop the production to find the source of the problem. The production teams then worked on a solution to prevent the same problem from happening again. This is a so-called Kaizen approach, to continually improve the workflow (Jones, Roos & Womack, 1990).

Toyota wanted their suppliers to be able to improve the quality of the parts they delivered. Toyota gave the suppliers specifications of what they expected of the product and the measurements and let the suppliers develop the desired product themselves. They encouraged the suppliers that were not in the same field to talk among each other to develop products that would work well together.

Toyota developed a just-in-time system for their production line called Kanban. The thought behind this system was that parts only should be produced to cover the immediate need of the next step. This is a so-called Lean approach to production, limiting the waste such as defective parts and other time wasting factors, and being able to see and prevent problems before they are happening. Another big factor in a Lean production system is to not produce more than just to cover the demands at that time. (Jones, Roos & Womack, 1990).

Lean production methods are used in several manufacturing companies two examples are Toyota and Sandvik. Based on Lean Thinking, Lean methods are now not only used in manufacturing they have also reached other fields like software development.
3.2.2 Lean thinking in software development
The best way to describe Lean thinking in software development is to understand the seven principles of Lean thinking. These principles are based on the work of Poppendieck (2003).

The first principle is to eliminate waste. It is important that the customer gets what he or she needs. Nothing should be developed that is not crucial or specifically requested by the customer. It is important to give the customer what they want as fast as possible. The customer has to get value for their investment at an early stage of the development.

Poppendieck (2003) presents some typical wastes of software development:

- paperwork – too much time is used to produce documentation
- partially done work – not all work is done
- extra features – unnecessary features that are not crucial
- building the wrong thing – lack of communication with the customer, resulting in a defective product
- waiting for information – waiting for information from the customer instead of contacting the customer and getting the information
- unnecessary functionality – developing functionality that is not going to be used by the customer, it is very common that customers actually use just a fragment of their software

The second principle is to increase feedback. It is crucial to have a working communication in Lean software development. The development team has an ongoing communication with the customer to be sure that they see things the same way. This prevents unnecessary functionality and helps to save time and resources for both the development team and the customers. It is also important to have a working communication within the development team and between the development team and the management. This will help to prevent bottlenecks in the development work. (Poppendieck, 2003).

The third principle is to defer commitment. In Lean software development it is important not to be tied up in commitments too early in the development work. As the development work is divided into segments it is no use to make commitments for a long term. There are also other factors that can change such as the customer’s business or the customer’s software situation. (Poppendieck, 2003).

The fourth principle is to deliver fast. It is important to deliver parts of the software as fast as possible to the customer. This will ensure that the customer gets what he or she wants and that none of the key functions are missing. Short iterations between deliveries to the customer will also make the total time of the development work shorter as the customer is able to decide to end the development work after any iteration. (Poppendieck, 2003).

The fifth principle is building in integrity. The customers overall experience is very important. It is important to keep the timeline for the segments and have working software at the end of each segment. All the parts of the software should work together seamlessly and the customer should feel like their money is well invested. The software should be well tested and fill the requirements and the development work should feel altogether professional. (Poppendieck, 2003).
The sixth principle is to empower the team. It is important to give the team the best conditions to succeed with the development. Poppendieck (2003) lists some of the key points for team commitment:

- a small team
- clear mission
- short timeframe
- staff with the necessary skills, both technological expertise and domain experience
- enough information to determine achievability
- freedom to make decisions
- basic environment for good programming

The seventh principle is to see the whole. As mentioned earlier it is very important that the different parts of the system work well together. It is also very important that the different parts interact well with each other. The consumer should feel like they have gotten a complete system that lives up to their requirements. (Poppendieck, 2003).

3.3 Kanban

According to Anderson (2010a) the Kanban method with its core properties gives organizations an adaptive system that will provide opportunities to get a Lean outcome. Here is a theoretical description of the Kanban method. For the practical description of the Kanban implementation process at Sandvik see 5.

3.3.1 The emergence of the Kanban method

The emergence of Kanban started in 2006 when Anderson (2010a) decided to implement a kanban pull system at the software engineering company he was presently working at. He was encouraged by the results and tried and developed the ideas that resulted in the Kanban method for technology and software development.

The word kan-ban, or kanban, spelled with a small k, comes from the Japanese word for visual card or signal card. The concept of kanban was first used in the Toyota manufacturing system. In a kanban system there are a fixed amount of cards (kanban) which corresponds to the decided capability of cards in the system. Each card represents one piece of work, and also acts as a signal. When a card is available the free card adds a piece of work which sticks with it through the whole system. If there are no available cards no more pieces of work can be started until another card is free. When a piece of work is done, the card is recycled and a new piece of work can be attached and put into the queue to go into the system. This kind of system is called a pull-system since new pieces of work are pulled into the system when they can be handled. Since new work is pulled into the system, its capacity cannot be overloaded if the number of cards has been correctly set. (Anderson, 2010a).

Andersons (2010a) Kanban method, spelled with a capital K, is evolutionary since it is based on the concept that changes should evolve when needed and should not be imposed. When changes are evolving in an evolutionary way and are adapted to the context specific process as they are in Kanban the resistance to change is minimal. The Kanban method starts the introduction of Lean ideas into Information Technology - IT, and software development. For more information on Lean see 3.2.

The repetitive work of the manufacturing process might seem like the complete antithesis of the knowledge work of software development. The software development process is highly variable, changes easily and is centered on soft valuables.
The manufacturing process is often the opposite, it has low variability, hard to change and is centered on hard valuables. Anderson (2010a) says that it is natural to be skeptical about how valuable Kanban can be in the IT work.

In Kanban it is important that the process is adapted to the teams way of working, Kanban is a tool to help visualize the process and encourages the tailoring of processes optimized for a specific context. This has been expressed in the slogan “Yes We Kanban”, in the meaning that in Kanban you have permission to develop a unique process optimized for your unique organization or unique team. (Anderson, 2010a).

3.3.2 The five core properties of a Kanban implementation
Anderson (2010a) there are five core properties of a successful Kanban implementation: visualize the workflow, limit work-in-progress, measure and manage flow, make process policies explicit and use models’ to recognize improvement opportunities.

The team visualizes their work by the use of a Kanban board see Figure 12, where the team presents all their work activities. This helps the team members to see what is going on at the time, what items are highly prioritized and also to show managers or customers their workload.

Another of the core properties of Kanban is to limit the WIP- Work In Progress. When there is a limit to the work that can be in progress at the same time it is easier to predict the time the work cycle takes. The quality of the work will also be higher because it creates slack time and more easily reveals where the bottlenecks are in the process when work is pulled into the system. (LimitedWIPSociety, 2009).

According to Anderson (2010a), by visualizing the workflow and limiting the WIP the team will get a more manageable workflow. It will be easier to see when new projects can be started and where there are bottlenecks or items that have been blocked for a long time. It is important that the team have policies that everybody has to follow. Among these policies you have the “definition of done” criteria or rules that have to be completed before the work item can be moved forward in the workflow. Other policies can include the color of the post-its for different types of work items and a timescale for the duration of the of different work items. There are different kinds of models and theories to use to further improve Kanban. These can be used after the team has worked with Kanban for a while and have knowledge and experience on the method. The models and theories can help with how to handle bottlenecks or to further improve WIP. (Anderson, 2010a).

3.3.3 Implementing Kanban
Anderson (2010a) gives his recipe for success for managers when to implement Kanban in their organization and the six steps in the recipe are to be followed in the order they come: focus on quality, reduce work-in-progress, deliver often, balance demand against throughput, prioritize and attack sources of variability to improve predictability.

Focus on quality means that the teams should be encouraged to produce initial high quality which will reduce function defects. This includes for example testing, code inspections, using design patterns and modern development tools. The focus on quality will result in a throughput improvement. After focusing on improving quality, it is time to reduce the quantity of work in progress.
Reducing the quantity of work in progress will result in a shorter average lead time, lead time is the time it takes a work item to pass through the work flow, and this causation is known in the manufacturing production as Little’s Law (Anderson, 2010a). Shorter lead time will also lead to better quality and the possibility to deliver more often.

Deliver often means more frequent releases of working code which builds trust with for example external teams and business sponsors, this event driven trust builds important social capital. Balancing demand against throughput means that work will be pulled into the system when other work is completed. New requirements can be accepted at the rate that other work is delivered. The limit of work in progress will then be set to a given size and the throughput of the process will only be constrained by a bottleneck. The team members in the bottleneck will be busy and others will experience slack capacity. This slack capacity will make the team members intuitively reduce the slack by improving their skills, tools, interactions and so forth, which will lead to the much-desired continuous improvement. (Anderson, 2010a).

Prioritizing means that it is time to turn attention to optimize the value delivered. This needs to be done after the ability to deliver and the predictability of delivery are secured. After building this level of maturity in the organization the last step of the recipe is to attack the sources of variability in order to improve predictability. Variability affects both the throughput and the value stream negatively. Small changes in for example process policies can give a much-desired predictability. (Anderson, 2010a).

Anderson (2010a) says that Kanban facilitates all the six steps in his recipe for success. When following this recipe different effects and behaviors will be visible in the organization.

### 3.3.4 Effects of Kanban

Here are some of the effects that Kanban have according to the literature regarding the relations with the customer, quality of work, task switching, lead times and satisfaction with work.

Regarding the relations with the customer according to Anderson (2010a) Kanban has shown to improve the satisfaction of the customer. This customer satisfaction comes from “regular, dependable, high-quality releases of valuable software” (p. 15, 2010). Kanban also helps to build trust with the stakeholders both upstream and downstream by providing the suppliers and the customers with a consistent and regular delivery pace. Kniberg and Skarin (2010) say that Kanban provides both the team members and the customers with important visibility to what effects their actions or inactions are causing.

Quality is a key aspect in the Kanban methodology. In Andersons (2010a) recipe for success for managers one of the steps is focus on quality.

Anderson (2010a) lists some steps to improve quality:

- “Code inspections improve quality.”
- “Collaborative analysis and design improve quality.”
- “The use of design patterns improves quality.”
- “The use of modern development tools improves quality.”
- “Reducing the quantity of design-in-progress boosts software quality.” (p. 24-25, 2010)

If the team have a focus on quality the effect will be an overall better quality. There will be fewer defects that will lead to less re-work. (Anderson, 2010a).
By limiting WIP the team have less work items in the workflow. This means that they have the ability to focus more on each and every work item. This leads to less switching between work items. It will also lead to better quality on the work items. Another advantage by limiting WIP is shorter lead times. Research on the subject says that with less work items in the workflow lead times will get shorter. (Anderson, 2010a). One of Anderson’s goals of Kanban is to improve employee satisfaction. By having a good balance between the team member’s work- and personal life, team members will be more motivated at work. More motivated team members lead to better performance in work.

3.4 Why choose Kanban?
Ladas (2008) expresses that Kanban is a practice that can be used to make a process that is specifically adapted to the problem. This means that the process will differ from the processes that other organizations need. Kanban will be specifically adapted to both the problem and the resources that are available. Ladas (2008, p.17) concludes with: “Kanban is a tool, and like any tool, it is meant to solve a problem. I think kanban solves this problem more efficiently than the known alternatives.”

Marschall (2010), co-editor of Agile Web Development & Operations, says that when choosing methodologies it is clear which to choose when: “If you already have working processes, which you want to improve over time without shaking up the whole system, Kanban should be your tool of choice.”

Anderson (2010b) says that: “If your organization has low maturity, limited capability at risk management, change management and decision making, and is riddled with specialization and defensiveness then Kanban is probably a better choice. If you have time to let the culture and performance evolve and improve over months and years then Kanban is the right choice.”

3.5 The connection between Software development methods, Kanban and Lean
There are several different views on how software development methods, Kanban and Lean are connected. Here are some views on how the connections can be seen.

According to Anderson (2010a) Kanban is the evolutionary change method that uses the visualization, the pull system and other tools to initiate Lean ideas into software development. Kanban is “applied to incrementally change the underlying process” (p.16, 2010a). This underlying process can be either agile or traditional software development methods.

Strickler (2009) expresses his view on how the Lean, Agile and Kanban are connected and how they differ, he says: “Lean and Agile are concepts that allow for more flexible, lower cost development or production – Kanban and Scrum are two approaches for implementing these concepts for software development.”

A founding member of the Lean Software & Systems Consortium, Cottmeyer (2011) says: “...agile is all about uncovering better ways of developing software by doing it and helping others do it. Lean/Kanban gives us another set of tools to help that come about.”

This shows that the views are not necessarily unison regarding the connections.
3.6 Our view of how Software development methods, Kanban and Lean are connected

In order for us to make our own understanding of the connections easier we made a graph of the connections as seen in Figure 5. This graph shows that the software development methods, either traditional methods or agile methods sometimes are insufficient and a need for change is requested. When a need for change is recognized the Kanban change method can be used to catalyze desired Lean behaviors.

The Kanban change method utilizes the Kanban system which uses the core properties; limit WIP, manage workflow, visualize workflow, make policies explicit and recognize improvement opportunities. When these core properties of Kanban are used they create the Lean behaviors; eliminating waste, increasing feedback, delaying commitment, delivering fast, building integrity in, empowering the team and seeing the whole.

This graph is our own interpretation of the connections between software development methods, Lean and Kanban.
4 Why Kanban at Sandvik?

This is the first empirical part of the report. It is a description of the problems identified at Sandvik and a description of the process in which Kanban was chosen to be implemented at SITS. The sources used for this part are company internal resources and interviews with staff at the SDO.

4.1 The Systems development Office – SDO

During the fall of 2009 Sandvik IT Services established a support office called Systems Development Office (SDO). The purpose of the SDO was to support and provide help to departments in the company in order for the departments to deliver solutions that have higher quality and to deliver them more effectively.

The expectations were clear; establish a common way of working that would raise maturity in software development throughout the whole company.

Johan Nordin at the SDO says: “Our mission is to give support to those departments that asks for our help. To ensure that our way of working would help the department mature we created a vision for software development. We asked ourselves if the work we were doing made us come closer or further away from the vision”. (personal communication, May 19, 2011).

The vision that the SDO created was: “To deliver exactly what the customer needs, when it’s needed, with total absence of waste in doing that – every time”.

The SDO established the vision with management as an incentive and as a guideline to what the office was working towards.

The SDO in the spring of 2011 consists of: Lotta Olsson the manager of the SDO, Johan Nordin the development support manager and Katarina Kvist senior systems developer who is on maternity leave since the fall of 2009. In the spring of 2010 Olle Thyrestam joined the group as in charge of testing. Christophe Achouiantz joined the group as Lean and Agile coach at the end of 2010 to help with the implementation of Kanban.

4.2 The concept of a common software development process

Johan Nordin describes that they were expected to create a common method of work. This would in a traditional way mean to introduce a new common work process, and make sure everybody was using it. Problems with creating a common work process were that they used different platforms within the company and there were old systems that needed to be maintained as well as a need to develop new systems. Johan Nordin says: “How was I supposed to find the right process to use, and how long would it take to gather experience from every department? How likely would it be that people embraced a work process if it interfered with the process they use today?” (personal communication, May 19, 2011). His conclusion was that they needed to create an alternative strategy.

The strategy was to develop a process for software development. The purpose with the process was to establish a framework and foundation of common terms. The idea was that if the SDO could find some common stages in the development process they could start describing what the software development teams were doing during those stages. This would make them able to communicate between different platforms without changing the way they worked.
The common method of work created is represented in Figure 6 as the systems development process.

![Systems Development Process Diagram](image)

**Figure 6. SITS systems development process**

The systems development process consists of three stages: solution design, develop feature and release to production.

### 4.2.1 Solution design
The solution design stage consists of understanding the needs, designing and planning. This stage is triggered by a request for a solution proposal and results in a solution proposal ready to be delivered to the customer.

The purpose of the solution design stage is to create a solution proposal that targets a specific business need, in order to provide more information to make a decision on how to proceed. The result should be on a level of detail that corresponds to the context and type of decision that is expected to derive from the result. The reason for grouping needs, solution design and planning together is to highlight that these activities often are highly dependent to each other and separating them often causes loss of tacit knowledge.

### 4.2.2 Develop feature
The develop feature stage consists of prepare for development, develop feature and verify with customer. This stage is triggered by a decision to start developing a feature and results in a feature that is ready to be released to production. In cases with more features than one that are developed for a software release, this process is executed several times.

Prepare for development is to make all necessary preparations required to get a feature to a state where it is ready for development. The purpose is that the development of the feature can be performed in a fast, flexible way without unnecessary delay or rework.

Develop feature is all activities needed to get the feature to a state where it is ready for production deployment. Common activities are; coding, creating and running tests, check-in to version control, update documentation etc. The purpose of this stage is being the stage where the real development of the feature is happening.

Verify with customer means to verify the developed feature with the customer (or other stakeholders such as end-users). The purpose is to get early feedback that what has been built is really what is expected. Making corrections at this stage is normally much more efficient than making them later in the process.
4.2.3 Release to production
The release to production stage consists of getting internal acceptance, getting customer acceptance and deployment. This stage is triggered by a decision to release the developed features to production and results in a new version of the system that is deployed in production and available to end-users.

Getting internal acceptance are the activities required to get to a state where the new version of the system is ready for customer acceptance testing. Different types of tests can be executed depending on the system. The purpose is to make sure that the software has the quality that is expected.

Getting customer acceptance is all the activities required to get the new version accepted by the customer. The purpose is to get feedback and acceptance from the customer to make sure that what has been developed really is what is expected. In some cases, this is completely handled by the customer and in some cases members from the development team can participate to help and support testing during this phase.

Deployment is all activities needed to get the release into production environment and make it available to the end users. The purpose is to make it possible for business to derive value from the new version that has been developed and that the deployment is performed without causing any failures.

4.3 The concept of Definition of Done
The assumption was that most of the teams within the organisation would be able to fit in rather well with the process they had described. When the stages were described on a fairly abstract level the focus was to identify where and when important decisions were taken.

The SDO established the term Definition of Done, DoD, which became the concept of what it means to be done in each stage of the process. The DoD would also help in raising quality when the team was united in what it meant to be done at each stage. The thought was that each team could describe its way of working against a common work process. Then the team would make a definition of done which would help them get good overview of what was done at what stage.

4.4 Workshops
Workshops were held with about 25 teams, and about a total of eighty persons. For each team during each occasion the results were documented. The SDO held the workshops in order to map out the teams. During the workshops the following steps were used:

1. Background and purpose; why are you here?
2. What is meant with software development? (from requirements to generated value)
3. Establishment of a value stream map for each team.
4. Answering a survey concerning how they work with software development more specifically.
5. Introduction to software development process (what and why).
6. Mapping of the value stream against the software development process.
7. Discussion on how work flows through the work process (waterfall, iterations, feature by feature or no control).
9. Listing of the five major problems the team has.
10. Work with cause and effect diagram for one of the problems.
This way the SDO could capture the present situation, value stream maps, cause and effects, and a first definition of done for a major part of the company. This generated a lot of knowledge for the SDO to be able to go further with, in the work process.

4.5 Problems in software development teams
During the workshops the SDO asked the teams to list their biggest problems. The SDO listed the issues and they were grouped into seven categories:

- Customer collaboration
- Technical debt
- Task switching
- Lead time
- Lack of competence
- Rework
- Quality and incidents

A major part of the issues were based around customer collaboration. The teams had issues with the requirements from the customer and they were not sure what the customer wanted and for that reason spent a lot of time trying to find someone with answers. It was also common that customers contacted the teams and pressured them to finish work. This in some cases led to so called “quick fixes”, working but not properly coded and/or tested software.

Another large category of the problems was technical debt. The most common issues in this category were the coding, the teams had no standards to follow and this led to unstructured programming. The teams also had issues with old and complex systems that would need to be modernized. These modernizations were not paid for by the customer and therefore were not done.

Task switching was another of the major issues. The team members had too many tasks going on at the same time and this created a lack of focus. Almost all team members had many simultaneous tasks going on and felt stressed that they did not have the time to do their work satisfactory. Team members were often switching between several tasks like support and development and then back again. Pressure from the customers to finish certain tasks also led to a lot of switching. The customers turned directly to the team members with change requests, i.e. a wish for changing something in an existing system, and the team members had problems saying no to them.

Long lead times were also a major issue, for example in some teams there was a two year queue for change requests and the overall lead time was too long, even relatively small changes made in a few days could take up to sixty days to go from idea to deploy.

The team members felt that there sometimes was a lack of competence, some teams lacked knowledge about the systems they were maintaining. There was a need for education concerning different systems, different tools and business knowledge. The unclear requirements or late changes from the customers led to a lot of rework and this rework took valuable time from the teams.

Many of these issues led to a lack of quality in the products, for example quick fixes, changes of requirements and task switching. This in turn led to unsatisfied customers and also to more rework for the teams. Generally most of the systems that had been around for a while had quite a large technical debt and a complexity that strongly influenced the possibility to further develop the systems.
During the workshops the conclusion was made that the problems were not that different between different development teams. Another conclusion was that most of the problems were connected with each other directly or indirectly, as seen in Figure 7.

### 4.6 Reasons for introducing Kanban

The main reasons for implementing Kanban at SITS were:

- Kanban addresses the root cause of the problems at SITS; Lots of work in progress. By limiting the work in progress the hope is that the other issues will also be influenced and a positive change will take place.
- Management made the decision to introduce the concept of Definition of Done, DoD. The DoD would be a tool to standardize and secure the quality of work in the teams.
- Visualization; one key ingredient to enable change is for the individuals in the teams and key stakeholders to understand the current situation. By visualizing the WIP and also to describe working policies it is possible to start focusing on the team’s current way of working, without putting any value into if it is a good or bad way of doing it, it just is...

Kanban is a tool that helps to manage the Work In Progress, WIP, the alternative tools in place within SITS before that was iterative methods like Scrum. But Scrum normally requires and introduces more changes from the beginning. Kanban is more of a way to help teams to evolve in their own pace. The teams themselves are in control over their own change and the pace that the changes are made.
5 Kanban at Sandvik

This is the second empirical part of the report. This is a description of the Kanban implementation process at SITS at Sandvik, starting with the preparations, then the Kick start and finally the Retrospectives. Since the Kanban method is meant to be adapted to suit the organization, this way of introducing Kanban is adapted to suit SITS. The sources used for this part are company internal resources, observations and interviews with staff at the SDO. This part of the report is one part of our mission at Sandvik; to make a documentation of the implementation process at SITS- Sandvik IT Services.

5.1 Kanban as a tool for improvement

This is a description and a documentation of the way that Kanban is on its way to be implemented at Sandvik IT Services. The sources used are company internal resources, interviews with staff at the SDO- System Development Office along with our observations during Kick starts and retrospective. According to the interviews with the staff at SDO we have described the SDO improvement concept, the Kanban coaching strategy and the operational excellence that goes via Kanban.

5.1.1 The SDO’s improvement concept

Figure 8 shows the improvement concept that the SDO coaches works through with a team. The goal of the concept is to identify improvement opportunities for each specific team and bring that team to improve itself.

![Figure 8. The SDO improvement concept](image)

A first contact is made where the SDO coach meets with the team leader or leaders and/or team manager or managers. The coach explains the improvement process and helps the team to formulate the team’s visions and goals. The wanted position for the team is clarified. The readiness and willingness of the team to change is discussed. The end of this phase is to get management approval for analysis. Then facts are gathered. The coach assesses the team using checklists and interviews. In this stage cooperation of managers, customers and other stakeholders might be required. After the fact gathering a desire for improvement is created by creating a common understanding of the current situation where problems are identified and discussed. This common ground is created in a meeting with the team, group leader, managers, and if appropriate also with customers and stakeholders.

The analysis stage consists of the coach analyzing the results of the assessments and the facts gathered. This stage might require further interviews with key persons. When this stage is done it should result in:

- known current situation
- known causes and effects
- a set of recommendations to address the current situation
Buy-in improvement plan stage means that the coach facilitates a meeting with the team, group leaders and managers. During this meeting the coach presents the results of the analysis and an improvement roadmap to address the current issues. The end of this phase is to get management approval for improvement.

Depending on the team’s maturity and context the coach most often recommends creating a baseline for improvements. Currently, the tool used for creating the improvement baseline is Kanban. Practically this stage is referred to as the Kick start, for more information on the Kick start see 5.2.

After an improvement baseline is in place, or when the coach assesses that the team has the necessary conditions to make successful improvements, the actual improve stage can start. This stage can have many forms in practice. It can be a series of workshops, a period of “live” daily coaching or one or several retrospectives. For more information regarding the Retrospectives see 5.3.

When the changes to bring about the envisioned improvements are executed the teams might need to reassess their work. This check is done with the coach’s help and may trigger a re-start of the process by gathering new facts and so on.

5.1.2 Kanban coaching strategy
The Systems Development Office- SDO is currently composed of four persons (two agile/lean coaches, one test coach and one manager) with the task to support and help to improve Sandvik IT Services development teams in Sweden. Development teams can also be found in approximately 40 countries, for example, the USA, Australia, Germany, South Africa, China and Russia.

SDO has therefore not the possibility to focus on only one team and support it on a daily basis. Focus is instead put on designing ways to help as many teams as possible by relying on key individuals within the team themselves.

The current coaching strategy can be formulated in this way:
- The SDO only supports teams that have explicitly requested help from the SDO and that the SDO has assessed to have a concrete desire for change and management commitment.
- Create a desire for change by, for example, the use of visualization elements for a team to see and understand their current situation.
- Provide support to the teams when their desire for change creates a concrete demand.
- Educate key individuals (Kanban Champions) within each team to promote Lean thinking and Kanban as well as handle daily meetings and flow management.
- Kick start teams to start a journey towards Kanban. Kick starts are whole day workshops where a SDO coach helps a team in creating the conditions for succeeding with Kanban.
- Support Kick started teams by participating in some of their daily meetings and giving feedback.
- Help the teams evaluating their position by facilitating regular retrospectives and ensure that they are on the right track.
- Introduce new Agile/Lean tools and methods on a just-in-time basis based on the results of the retrospectives.

Using this strategy, SDO is currently supporting 20 teams of 5 to 12 persons located in Sweden. Another 10 teams are requesting support in their quest for improvement. Many other teams are also in early discussion with the SDO, even teams that are not located in Sweden (e.g. Brazil).
5.1.3 Excellence goes via Kanban

Operational excellence is achieved by improving continuously, which is made possible by Kanban. The roadmap towards excellence is therefore also a roadmap towards Kanban, but only as a means to an end. Sandvik’s roadmap towards operational excellence consists of four concrete steps: create a baseline, establish flow, create capacity for improvements and optimize, as seen in Figure 9.

![Sandvik’s roadmap towards Kanban](image)

The first step is to create a baseline for improvements. This baseline is created by the team discussing, agreeing and defining their current way of working. A baseline is absolutely necessary to secure that all changes that will be made to the team will be sustainable. The SDO’s experience shows that forcing or hurrying changes on a team that has not agreed on a common way of working will simply not stick. Sooner than later, the team will reject the changes and the effort spent by SDO would be wasted. A baseline includes both the process the team uses, i.e. their work process and their definition of done, but also the rules that will govern their work, i.e. their policies. Once created the baseline must be known and followed by all team members. The baseline must be kept alive by continuously adapting it to the team’s actual policies and rules. The most important step in creating a baseline is to see and understand the team’s current way of working (their workflow) and define the exit criteria for each step on the workflow. This is all done during the Kick start.

The second step is to measure and manage the work flow. The first goal is to establish flow, then measure it and finally optimizing it. One of the first steps in establishing flow is focusing on moving smaller units of work through the entire workflow instead of big batches of work being moved in different phases through handoffs. The flow must then be managed to ensure that there is a rather constant input of work and that the work flows as smoothly as possible until completed.

The third step is to create capacity for improvements by limiting the work in progress. In SDO’s experience this crucial step in enabling a Kanban system has most chances of sticking with a team when introduced after having established flow. This is mainly due to that only then has a team matured enough to understand and to maintain the limit WIP policy.
Finally a team is ready to take to the last step towards Kanban, and the first towards reaching operation excellence; Optimize. Only then has a team the tools to both see and understand its limitations, the necessary maturity and/or courage to tackle the problems and the experience to sustain that “continuous improvement” state-of-mind. Depending on the team’s maturity and its context, getting to the Optimize can take the teams months or maybe years. The team focuses on optimizing its workflow. The goal is a continuous, steady, even and smooth flow of value from request to delivery in the hands of the customer.

This pull strategy reduces the risk of a wasteful over production and also makes it easier to discover problems. Examples of waste can be unnecessary handoffs, bad specifications, defects, rework and unavailable resources. Every problem will make the flow slow down or make the flow uneven.

5.2 The Kick start

The Kick start is the first practical step for the teams in implementing Kanban. The main goal with the Kick start is that the team will improve from its own individual starting point to the point when the team is ready to start implementing Kanban the very next day. When the Kick start is done the team has a Kanban board, the team’s principles and the team’s Work In Progress, WIP.

Henrik Kniberg helped the SDO to review their setup for the Kick start. The Kick start starts with a description of the goals to be reached when fully implementing Kanban. The goals are that the team feels like a team and that the team members feel that they are in control over their work. The teams will be sure that they are focusing on doing the right things and deliver value to their stakeholders. The stakeholders are the partners both downstream and upstream that have interest in the team’s work. The goals are also that the persons in the teams have time to improve both themselves and their team and not feeling stressed though they have much to do.

The teams are also briefed on the topics of flow, push- and pull system. The flow is the flow or streaming of work that should not be interrupted. A push system pushes work into the work process of the teams whether the team has capacity or not to handle the work at that time. The pull system means that the team itself pulls work into the team’s work process when the team is ready and has capacity to handle the work.

The goal of the Kick start is for the teams to agree and define a way to visualize their current reality. They will also agree and define their current way of working; their work processes, their definitions of done and their work governing policies.

An important factor of making the first process as smooth as possible is to; “When in doubt, keep it simple”. This was an important advice given by Henrik Kniberg to the SDO when making the preparations for the first Kick start. During the Kick start the team does not have unlimited amounts of time to discuss details, consequently it is better to leave detailed and time consuming discussions and decisions for later on. Everything in the process can be negotiated and renegotiated whenever the team feels a need for it after the Kick start.

To make sure that the visualization will help the team solve its most important problems the teams perform a short retrospective; the team members individually write down on post-it-notes the top three problems that they currently have either individually or as a team. All the problem notes are gathered on the white board and then grouped into categories to map up the top three most common problems for the team.
The team also writes down their policies and rules discussed during the whole Kick start and gather them on a Policies flip-chart, seen in Figure 10. These policies are the rules that will govern the teams work. The policies are created by the team and for the team.

![Policies Flip Chart](image)

**Figure 10. A team’s policy flip chart**

The team will also need to discuss and define who is on the team since the goal is to visualize all the work that is handled by all the team members. Sometimes there might be team members that divide their time between different teams and therefore are not present during the Kick start. The team will also need to discuss who their stakeholders are. They need to define who their upstream partners are, i.e. their providers and also their downstream partners, i.e. their customers.

### 5.2.1 Work types

When the team knows where their work flow starts they need to identify what types of work that arrive at that point and what other types of work that exists within the workflow. The work types are groups of work items that have similar properties and mostly behave in the same way. Different kinds of work types might have for example different priorities, behaviors, focus or goals.

The team will then investigate what types of work items they have in their own work flow. Then they need to sort the work items into different types or groups. The team will then select the work types that they would like to visualize on their Kanban board.
For the visualization effect it is best not to try to have more than five different types of work on the Kanban board since it might be hard to get an overview if the Kanban board is too busy. The team should prioritize the work types that take most of the groups’ time and use these as their work types. Typical work types can be: incidents, change requests, project work, pre-studies, defects, documentation or improvements.

The team chooses a post-it-color for each decided work type the colors available at SITS are yellow, bright-yellow, green, orange and cherry. The team should not use the pink if not absolutely necessary since that is the color reserved for blocked items. The small pink post-it is reserved for marking blocked work items. The cause of the block should be written on the small pink post-it and stuck on the blocked item. It can then be removed when the work item is no longer blocked. The chosen colors for the work types need to be documented on the Policies flip-chart, see Figure 10, the team should write the name of the work type on the selected post-it-note, for example incidents on the bright-yellow ones, defects on the green and so on.

### 5.2.2 Work items

When colors are chosen for the different work types it is time to visualize the work items for each work type. The team should discuss and agree on a work item template. When visualizing each work item the goal is to have just enough information on each post-it to be able to make good decisions when standing in front of the Kanban board. The team should avoid overloading each work item with information. The work item template needs to be documented on the Policies flip-chart.

![Figure 11. Description of a team's work item](image)

As seen in Figure 11, at the top left corner is the “hard” date, the final date of the work item, if needed. Under the “hard” date is the E-ticket ID, if available. The E-ticket ID is the identification number of the work item in Sandvik’s intranet. At the top right corner is the size estimate. The size estimate can be divided into sizes; XS < 1 day, S 1-2 days, M 3-5 days, L 1-2 weeks and XL 3 weeks +, for example a S work item is a piece of work that will take one team member one to two days to complete. These sizes can be adjusted to fit the team’s size demands. At the bottom left corner is the date when the work item is started, i.e. pulled in from the next column on the Kanban board as seen in Figure 12. At the bottom in the middle are the initials of the team member or members that have worked on the item. At the bottom right corner is the end date, i.e. the date which is the actual done date of the work item. In the middle of the work item the title is put and if needed some description can be added.

The work types can be compared to the top three common problems that the group has. This can be used to reflect on and to highlight if issues are connected to certain work types.
5.2.3 The next and done columns
The flow in the process goes from next to doing to done. Work items are first put into the next column. The next column is an ordered list of work items. The first item at the top in the next column is the next one to be worked on. Maximum of items in the next column should be 10-15. The next list can be changed and reordered at any moment by the customer or his/her representative in the team. The doing stage is divided into several sub stages. The done column is for the work items when no more work is required on them, i.e. they are really done. The done list can be emptied at any moment by the customer or his/her representative in the team. Some teams have celebrations when they have reached a number of work items in their done column.

5.2.4 The systems development process
In the first Kick starts that were held the groups themselves described their own work process, this part took about two to three hours to complete in a good way which the coaches felt was too long time. After these first Kick starts the coaches decided to use a work process description with the SITS systems development process in mind, for more information on the systems development process see 4.2. The teams were allowed to change the work process description to fit the teams own work process. The whole process of defining the work process was sped up this way but the teams still gained the same results. It was also natural to use more common terminology between the teams at SITS.

<table>
<thead>
<tr>
<th>Develop Feature</th>
<th>Release to Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>Prepare for dev.</td>
</tr>
<tr>
<td></td>
<td>Dev. Feature</td>
</tr>
<tr>
<td></td>
<td>Verify with Customer</td>
</tr>
<tr>
<td></td>
<td>Get Internal Acceptance</td>
</tr>
<tr>
<td></td>
<td>Get Customer Acceptance</td>
</tr>
<tr>
<td></td>
<td>Deploy</td>
</tr>
<tr>
<td></td>
<td>Done</td>
</tr>
</tbody>
</table>

As seen in Figure 12 the same flow of the SITS systems development process is represented on the Kanban board. The stages of the teams actual work process should be the same stages that are represented on the team’s board.

5.2.5 Defining the teams own definition of done
Each stage of the work process must have clear exit criteria that define when a work item is ready to enter the next stage. The definition of done is the collection of all exit criteria for each stage. The definition of done is a tool for the team to establish and maintain quality. Sometimes criteria in the definition of done should be put as column, and sometimes a column can be put as a criteria in the definition of done.
That definition is up to the team to decide, when a definition of done is very important maybe it needs to be put on the board as a column in order to be highlighted.

When the team knows the concept of the definition of done they should process, discuss and define the team’s own definition of done. The definition should be formulated as the exit criteria from each column focusing on the results that need to be attained in order to move a work item from one column to the next column. The exit criteria will be placed at the bottom of each stage of the team’s work process. The teams should reuse the SITS terminology as far as possible, if needed they can remove or add criteria from the SITS standard. After creating the teams own definitions of done then the team should try the definitions by using several real life work items examples to see that the definitions of done are working with actual work items.

<table>
<thead>
<tr>
<th>Develop Feature</th>
<th>Release to Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>Prepare for dev</td>
</tr>
<tr>
<td></td>
<td>Dev. Feature</td>
</tr>
<tr>
<td></td>
<td>Verify with Customer</td>
</tr>
<tr>
<td></td>
<td>Get Internal Acceptance</td>
</tr>
<tr>
<td></td>
<td>Get Customer Acceptance</td>
</tr>
<tr>
<td></td>
<td>Deploy</td>
</tr>
<tr>
<td></td>
<td>Done</td>
</tr>
</tbody>
</table>

Figure 13. A Kanban board with the definitions of done at the bottom

After establishing the definitions of done and placing those at the bottom of the Kanban board it can look like the one in Figure 13.

5.2.6 Defining the teams work process and populating the Kanban board
At this stage the team should process, discuss and define the team’s own work process. The work process must fit all the work types that the team has selected. The team creates a first draft on the board with columns preferably inspired by the SITS process and reuses the standard SITS process names as far as possible, if needed the team can remove or add steps from the SITS standard. After creating the team’s own Kanban board adapted to the team’s work process then the team should try the process using several real life work items examples to see that the work process is working with actual work items.

When the work item templates, definitions of done and the board all are satisfactory it is time for the team to populate the Kanban board. The team should now create work items for all ongoing work. When doing this the teams should use their work types and their work item templates. When placing the work items on the Kanban board the team should check the definitions of done for knowing where to place each work item.
When the team’s first complete version of the Kanban board is done it is time for the teams to reflect on the Kanban board. What thoughts are triggered when looking at the Kanban board? The team can be helped by using these questions:

- Is there a stage in the process that is overloaded? Why?
- Is there a stage in the process that is empty? Why?
- Is there a dominant work type? Why? Is that in line with the goals?
- Is there a rare work type? Why? Is that OK?
- Are the top problems or issues of the team visible?

The team is coached to reflect on the Kanban board themselves and to continue the reflections after the Kick start.

When using the Kanban board some effects can be valuable to improve the board with. Not all teams might find all the tips and tricks valuable, the most important thing is to make the Kanban board valuable for the team, i.e. using the tips that are valuable for just that team.

Avatars can be used to highlight the work item or items that each person is working on that day. The avatar consists of a magnetic button that each team member writes his or hers initials on. The avatar is every day placed on the work item post-it that is the person’s work item for the day, see Figure 14. This makes it visible on the board what everyone in the team is doing that day. According to where the avatars are placed it is also possible to see where in the work flow the main focus of the team is.

It is also possible to divide each column of the work flow into two parts, see Figure 14. For example the column prepare for development can be divided into doing and done. When working on the preparations of a work item for development the work item is in the doing column and when it is done it is put into the done column. The work items then sit in the done column until they are pulled into the next column. This done column is a signal downstream that there is a work item ready to be worked on. This way of dividing columns can be a powerful way to visualize the bottlenecks in the work flow. Where there are queues there might be a bottleneck to work on.

Impediments can also be visualized by using a visual signal for a blocked work item. This visual signal is usually a bright pink smaller post-it note that sticks to the bigger work item post-it. On this blocked work note the reason for the block is noted for example waiting for customer, and also a progress bar that enables the team to make a mark each day a work item is blocked.

Creating buffers in the Kanban board can identify work items that are out of the teams control and therefore not actionable. For example after a feature is developed a “wait for release”-column can be created. Another useful column can be “wait for deploy”.

Priorities can be visualized with urgent-markers, often used are red magnetic buttons. These red buttons are marking work items that are urgent to expedite. When there is an urgent marker on the Kanban board the whole team drops everything that they are currently working on and swarms on the urgent work item until it is done. This means that all other work items are put on hold. Since this is a powerful tool and puts the other work on hold it should be used restrictively and used maximum one at a time.
For urgent work the board can be organized with a separate urgent field across the whole board as an urgent swim lane. In this urgent swim lane the top prioritized work are placed as seen in Figure 14.

![Figure 14. Visualizing priorities with swim lanes and using avatars](image)

If desired the whole Kanban board can be set up with swim lanes for prioritizing work. The swim lanes can be divided into low, standard and urgent. When distributing the work in the team the urgent swim lane is prioritized, then the standard and lowest priority is the low lane. Sometimes it can be favorable for a team to for example decide that every 10th work item should be pulled from the low lane. In the low lane there can be work items that though they are not highly prioritized still can be valuable to some of the standard or urgent prioritized work items.

In managing the next queue with new incoming work items there can be created a column for incoming items and items in the backlog. The incoming items are the ones prioritized after each daily meeting to be pulled into the next column. The remaining work items that are not prioritized are put back in the backlog until the next daily meeting when a new prioritization is done.

5.2.7 Daily meetings
The daily meetings are intended for the team to reflect on their work. In order for this to be possible it is important that the board is up to date and reliable. The Kanban board should be updated every day either before or during the daily meeting if that is not possible it should at least be updated as soon as possible. The team meets in front of the board to discuss and take actions on problems, risks and opportunities. The daily meeting is not supposed to be a status meeting the Kanban board should already have made the status visible. The focus should be on what is going to be done not what has already been done.

The whole team should be present at the status meeting, and the meeting should be a stand-up meeting in front of the Kanban board. The maximum time spent on the meeting should be 15 minutes. One person of the team is the facilitator of the stand-up meeting and this task can preferably be rotated among the team members. Focus on the meeting should be on the flow of the work. The facilitator “walks the board” from right to left focusing on the work items.
Together the team identifies where help is needed, where there are blocked items and bottlenecks. Discussions of how to solve problems should be kept brief. If deeper discussions are required the discussions should be postponed to an after meeting with the team members that are concerned.

The team should agree on where and when the daily meetings will take place and also agree on who will facilitate the meetings. The agreements concerning how to hold the daily meetings need to be documented on the Policies flip-chart.

At the first daily meeting it is important to discuss what changes need to be made to the Kanban board to facilitate the discussions.

5.2.8 The end of the Kick start
When the team has been Kick started the team has created a baseline. They have visualized the workflow with the help of the Kanban board. The team has also made their process policies explicit with their Policies flip-chart. The team is now in control and the board reflects their way of working. The team itself is best placed to find a way of working that fits their team and their context. The Kanban board is mirroring how the team is doing. The goal is for the individuals and the team to improve.

In visualizing the work on a Kanban board it might highlight the problems of the team. The team and others can sometimes see and understand the problems just by looking at the board. It is also easier to address the problems when you can define an effective visualization scheme. The team works on the right things and team members help each other. The team spirit grows stronger as the teams can articulate the way they work together. To see pictures of actual Kanban boards at SITS see appendix 2.

The existing roles, responsibilities, job titles and practices are kept intact. The only changes should be with the interface upstream and downstream, like business owners and systems operations. This means that initially the change will be in optimizing the process and then it will gradually mature into a capability that leads to larger managed changes.

5.3 The Retrospectives
The Retrospectives are regularly recurring workshops with the team, and in the beginning facilitated by the coach. The intent of the first two retrospectives, two weeks apart, are to evaluate how the work from the Kick start is progressing and how the team feels about it.

In its simplest form, the workshop is held like this; every member of the team gets a number of post-it notes to write down things that in their opinion, has become better and/or worse since the Kick start. When the group is done with the writing each of the team members present to the others what they have written. The team members stick the post-it to a whiteboard with two columns, one for better and one for worse, and explain their opinions. This Retrospective board can be seen in Figure 15. When all post-its have been posted on the Retrospective board the team discusses how to solve the issues in the worse column. The group also have the opportunity to discuss how to solve problems that are not connected with the Kick start or Kanban.

The coach will help the teams in seeing and understanding current problems and the coach will help the teams in resolving their problems.
The Retrospectives are the main source of information for the coaches to assess how the teams are progressing on the way of implementing Kanban. It also helps the coaches to find out if and how the implementation process can be improved.

In the Retrospectives the teams are starting the work of improving themselves. This part of the process is really a pull process, new Agile and Lean tools are pulled in by the team when the team is ready for them. The changes are made at the rate that the team can cope with. After the Kick start, the teams have at least three Retrospectives that are facilitated by the SDO coach. Further Retrospectives continue in the same way, though they may be facilitated by someone else, like the team’s Kanban Champion.

5.4 Our summary of Kanban at Sandvik

Seen in Figure 16 is our summary of the beginning of the implementation process of Kanban at Sandvik.

The improvement concept is the process that the coach works through with the teams. In this process the coach assesses and analyses the team’s wants and goals with the team’s work process. The Kick start is the first step for the teams on the way to implementing Kanban. The Kick start consists of defining policies and creating and populating the Kanban board. The next step is the Retrospectives where the teams evaluate how their work is progressing.
6 The effects of Kanban at Sandvik

This part of the report is about the effects that Kanban has had at SITS at Sandvik. The software development teams themselves are measuring hard data and collect statistics that have easier measured effects. We have focused on the softer data that is more problematic for the teams themselves to measure and collect statistics from. Here we present the results from the answers of the survey interviews with 15 of the team members in the software development teams that use Kanban at SITS. In the tables we have cited the most vigorous statements. When a panel is empty there is no answer in that category from the interviewees. The answers are divided into time categories depending on how long ago the teams started the Kanban implementation. The time categories are 2 months or less, 3-5 months and 6 months and more. There was a relatively even distribution of the teams in the time categories, five teams in the less than 2 month’s category, six teams in the 3-5 month’s category and four teams in category for 6 months and more. The time categories are important in order to see if there are effects that are developing over time.
6.1 Effects on task switching

Table 2 shows the most vigorous statements that the interviewees have made regarding the effects that the Kanban implementation has had on task switching.

<table>
<thead>
<tr>
<th>Positive effects</th>
<th>Negative or no effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prioritization of work</td>
<td>Work process</td>
</tr>
<tr>
<td>&gt;6 months</td>
<td>“Prioritization of work items”</td>
</tr>
<tr>
<td></td>
<td>“Prioritization better... a better overview of upcoming tasks and how to distribute them in the team.”</td>
</tr>
<tr>
<td>3-5 months</td>
<td>“Easier to see prioritization of ongoing work, lower down-&gt; lower priority”</td>
</tr>
<tr>
<td></td>
<td>“We are used to prioritization since Scrum and we have continued in the same way.”</td>
</tr>
<tr>
<td>&lt;2 months</td>
<td>“Good forum to be able to prioritize together”</td>
</tr>
<tr>
<td></td>
<td>“More even flow”</td>
</tr>
</tbody>
</table>

Table 2. The effects on task switching

Several of the team members feel positive effects on the change in prioritization of work items, both regarding prioritization of ongoing work and distributing of work within the team.

Something that also has been majorly affected is the work process. The team members feel that it is easier now when they do not have as many ongoing work items at the same time and the focus has shifted from starting new work to pulling work items through and completing work instead of pulling new work in. This pull process is focused on by several of the interviewees and they feel that the flow of work is more even now.

One team member says that Kanban feels like a disruption with the work. Another problem is that some of the team members feel that problems are not solved but made visible and evident with the Kanban board like for example problems with blocked items. Another problem is that team members are still switching between work items.
6.2 Effects on technical debt, quality and incidents, and rework

Table 3 shows the most vigorous statements that the interviewees have made regarding the effects that the Kanban implementation has had on technical debt, quality and incidents, and rework. Quality and incidents, technical debt and rework are all closely related to each other that is why we have chosen to display them together.

<table>
<thead>
<tr>
<th>Positive effects</th>
<th>Negative or no effects</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Focus on quality</strong></td>
<td><strong>Cooperation</strong></td>
</tr>
<tr>
<td><strong>&gt;6 months</strong></td>
<td></td>
</tr>
<tr>
<td>“...we are gaining a better understanding of how to work which step by step leads to better quality.”</td>
<td></td>
</tr>
<tr>
<td>“...some defects have been corrected faster since the implementation of Kanban.”</td>
<td></td>
</tr>
<tr>
<td>“...we discuss and work more with quality issues now than before.”</td>
<td></td>
</tr>
<tr>
<td><strong>3 -5 months</strong></td>
<td></td>
</tr>
<tr>
<td>“We discuss and work more with quality questions than we did before”</td>
<td>“Easier to ask for code review and help when we meet and talk for a while each morning”</td>
</tr>
<tr>
<td>“We have more focus on requirements now...”</td>
<td>“Since we have better understanding of our colleagues work, we can help each other with things they might not have thought about.”</td>
</tr>
<tr>
<td>“We have columns for quality control.”</td>
<td>“The resources check each other’s coding...”</td>
</tr>
<tr>
<td></td>
<td>“Higher quality in the long run when we can discuss different applications complexity, who knows what, who should, can we transfer competence and who can support...”</td>
</tr>
<tr>
<td></td>
<td>“… we are beginning to use code review but it takes a while to implement”</td>
</tr>
<tr>
<td><strong>&lt;2 months</strong></td>
<td></td>
</tr>
<tr>
<td>“... we do not forget work items”</td>
<td>“No change for the better, no change for the worse.”</td>
</tr>
<tr>
<td></td>
<td>“More discussions concerning consequences, more clear cut discussions”</td>
</tr>
</tbody>
</table>

Table 3. The effects on technical debt, quality and incidents, and rework

Several of the team members feel that they have more focus on quality than before and they also discuss quality issues and consequences now more than they did before. Defects are being corrected faster. The teams have special columns for quality control and discuss and work with quality issues. Some teams also have more focus on requirements now.

There are a few team members that have not seen any changes in quality, but some team members hope for better code quality.

Team members think that it is easier to get help from colleagues, both when asked for and when a colleague might have ideas for improvements. More focus is put on who should do what and on helping each other.
6.3 Effects on customer collaboration

Table 4 shows the most vigorous statements that the interviewees have made regarding the effects that the Kanban implementation has had on customer collaboration.

<table>
<thead>
<tr>
<th>Positive effects</th>
<th>Negative or no effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer prioritization</td>
<td>Customer understanding</td>
</tr>
<tr>
<td>&gt;6 months</td>
<td>“It is easier to visualize the prioritization to the customers in front of the Kanban board…”</td>
</tr>
<tr>
<td>3-5 months</td>
<td>“Our customers are participating in the prioritization process. Every two weeks the customers get a follow up and discussion on the status of the board.”</td>
</tr>
<tr>
<td>&lt;2 months</td>
<td>“We have shown the customer the board and they are positive…”</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. The effects on customer organization

For some team members it has positively affected the customers understanding of the work process. The team members feel that the customers understand the team’s way of working and what items to work with and how long time it takes to work on some work items.

Team members also think that the customer is more active in the prioritization process and the reprioritization. They also feel that the customers are making the priorities visible for the team and this also makes the customer sometimes see when work items are blocked when waiting for the customer’s own approval. The responsibility allocation and participation of the customers in the work process are also better than before.
Some team members feel that they have had no effect on the customer collaboration compared to before.

6.4 Effects on the time aspect

Table 5 shows the most vigorous statements that the interviewees have made regarding the effects that the Kanban implementation has had on time.

<table>
<thead>
<tr>
<th>Positive effects</th>
<th>Negative or no effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus on collaboration to improve workflow</td>
<td>Changes in lead time</td>
</tr>
<tr>
<td>Better or the same</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Duration</th>
<th>Interviewee Comments</th>
</tr>
</thead>
</table>
| >6 months | “We have to improve cooperation within the team to shorten lead times.”  
“It is easier to keep focus on what is important.” |
| | “It feels rather positive that we are not measuring the lead times as much as we did before.”  
“One can see what workload each person has so it is easier to time plan work items” |
| | “We have fewer long hauls” |
| | “The team recognizes their problems to get work done. Which makes us cooperate to pull through a work item” |
| | “We feel calmer when we do not have to worry about the time aspect all the time.”  
“We used Scrum but switched to Kanban in order to get rid of time based planning (sprints)”  
“The team is more satisfied with the work. By not having to make sprint planning…” |
| | “I feel like our work items have shorter lead times, but there is still a lot of blocked items.”  
“Becomes more structured and therefore probably quicker lead time” |
| <2 months | “We have shorter lead times when everybody works on the same work item”  
“More collaboration around deadlines”  
“More structured work and therefore probably shorter lead times” |
| | “…focus is now on lead time.” |
| | “All work items gets a 1 hour and 15 minutes delay per week”  
“We still have to long lead times… I expect we will shorten lead times when we take the next step and start limiting our work in progress.” |
| | “We have shorter lead times when everybody works on the same work item”  
“More collaboration around deadlines”  
“More structured work and therefore probably shorter lead times” |

Table 5. The effects on the time aspect

Some of the team members feel that they cooperate more in order to pull work items through the workflow faster. While other teams feel like they have to improve their cooperation even more to be able to shorten their lead times.

Blocked items are still a big problem that keeps the lead times longer than they have to be. The lead times are somewhat shorter now than before and there is more focus on what has to be done.

Some team members feel that not having to do sprint planning is better now than before when they were using Scrum. Some feel that time planning is easier now when everybody’s workload is visible.
One team member thinks that lead times will be shorter when they start limiting work in progress. Another team member thinks that the morning meetings create a delay in the work process.

### 6.5 Effects on satisfaction

Table 6 shows the most vigorous statements that the interviewees have made regarding the effects that the Kanban implementation has had on satisfaction.

<table>
<thead>
<tr>
<th>Positive effects</th>
<th>Negative or no effects</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Focus on collaboration</strong></td>
<td><strong>Visibility</strong></td>
</tr>
<tr>
<td><strong>&gt;6 months</strong></td>
<td>&quot;Better cooperation within the team&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;YES, most of all we distribute the work load by the board and have better insights on what everybody does so that we can distribute the work equally&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;We can discuss problems easily and quickly by the board which gives both increased comfort and productivity&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Most problems are made visible&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;The stress is on the group instead of on the individual when there is a lot to do&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;The mission is made visible.&quot;</td>
</tr>
<tr>
<td><strong>3-5 months</strong></td>
<td>&quot;We are all more visible for each other (we used to sit with our own systems and worked all by ourselves) we work as a team now&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;We have a better overview... and that is very nice for me. We also have that time in the morning to air our problems and help out. Not all of us are easily asking others for help and now the focus is on that&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Yes it is better when it is clearly visible what everybody works on&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;I think it is easier to see what is being done and who is doing it...&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Easier to discuss problems and solutions.&quot;</td>
</tr>
<tr>
<td><strong>&lt;2 months</strong></td>
<td>&quot;Easier to get an overview of the work flow and not as much work in progress...&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Better cooperation&quot;</td>
</tr>
</tbody>
</table>

---

41
Team members say that they feel less stressed when they do not have to worry about the time aspect as much as before.

They also experience now that they have a better overview on what is going on and what the other team members are working on. This make them feel more like a team. It also makes the cooperation within the team better. The focus has shifted to collaboration and cooperation.

Something that also has been positively influenced is the ability to get help with problems. Team members feel like it is easier to ask for help than it was before. An important factor is the morning meetings where the team members have an opportunity to discuss problems and help each other to solve them.

Some of the team members say that they had expected more of the Kanban implementation, but they express that it might take a while for the effects to be visible. One of the teams still needs a strong leadership and they do not work as independently as intended. There are also team members that think that Kanban is just a waste of time and that it disrupts the daily work. It can also be discouraging when large work items do not seem to move on the Kanban board.

Several team members think that they have a better overview of what everybody in the team is doing and they feel that it is more visible now what each team member is doing.

Several of the team members also experience that it is easier to work together now. They have a more shared responsibility and feel that it is easier to discuss problems and solutions. Several feel less stress and they say that it has not made them go down in efficiency rather the opposite.
### 6.6 Our summary of the effects of Kanban at Sandvik

This is our summary table of the most important positive and negative effects of Kanban at Sandvik it is seen in Table 7.

<table>
<thead>
<tr>
<th>Positive effects</th>
<th>Negative effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Better prioritization of work</td>
<td>• Not all problems are solved with Kanban</td>
</tr>
<tr>
<td>• Less task switching</td>
<td>• Some team members had higher expectations than are met (yet)</td>
</tr>
<tr>
<td>• More focus on quality</td>
<td>• Problems are made visible but solutions are not always available</td>
</tr>
<tr>
<td>• Better cooperation within the teams</td>
<td></td>
</tr>
<tr>
<td>• Better collaboration with the customers</td>
<td></td>
</tr>
<tr>
<td>• The customers are more involved</td>
<td></td>
</tr>
<tr>
<td>• Somewhat shorter lead times</td>
<td></td>
</tr>
<tr>
<td>• Less stress in the teams</td>
<td></td>
</tr>
<tr>
<td>• Work flow more visible for the team</td>
<td></td>
</tr>
</tbody>
</table>

Table 7. Summary of the effects of Kanban at Sandvik

The major positive effects are that the teams feel that they have better prioritization of work and a little less task switching now. The teams have more focus on quality and they feel that they have better cooperation within the teams. They also feel that they have better collaboration with the customers and that the customers are more involved now. They also experience somewhat shorter lead times now. The stress is less on the individual team members and the work flow is more visible for the team.

The major negative effects the teams feel are that not all problems are solved with Kanban. Some team members had higher expectations than what are met yet. They also express that problems are made visible but solutions are not always available in Kanban.
7 A comparison of the theoretical effects and the actual effects of Kanban at Sandvik

This part is a comparison of the empirical data and the theoretical frame of reference. We will start this by going back to the research question that we started with. The question was:

What effects have the Kanban implementation had on the problems in the software development teams?

7.1 The effects according to the theory

When applying the five core properties of Kanban it will result in effects. The core properties are; visualize the workflow, limit work-in-progress, measure and manage flow, make process policies explicit and use models’ to recognize improvement opportunities, for a more extensive explanation on Kanban see 3.3.

Effects will be evident even though the five core properties are not fully implemented. For example with making process policies explicit, effects on the work are happening when the first policies are worked on even though not all the team’s policies are made explicit yet. Also for example working towards limiting work in progress will have effects on the work even though an implementation of limiting work in progress is not fully done yet. The same applies to the other three properties; visualize the workflow, measure and manage flow and use models’ to recognize improvement opportunities. During the process of working with the properties effects are evident.

7.2 The problems at Sandvik

These were the problem categories that SITS felt were the top ones. Read more about the issues the work teams had before the implementation of Kanban in 4.5. A summary of the problem categories are;

- Task switching
- Customer collaboration
- Lead time
- Quality and incidents
- Technical debt
- Rework
- Lack of competence

7.3 The effects reached with Kanban at Sandvik

This is our display- and analysis model for displaying the results of our analysis. The analysis is based on the answers from our survey interviews with the team members of the software development teams at SITS.
The degree of outcome is divided into four categories:

- Not supported - the outcome has not reached the expected influences of Kanban or is not supported at all by Kanban.
- Beginning to be supported - the outcome at SITS is on its way towards reaching the expected influences of Kanban.
- Well on its way of being supported – the outcome at SITS has almost fully reached the expected influences of Kanban.
- Fully supported – the outcome at SITS has fully reached the expected influences of Kanban.

Table 8 is displaying the problem categories the work teams had before the implementation of Kanban, the evaluation of the problem categories after using Kanban for up to two months, the expected influences of Kanban according to the theory and to what degree the outcome is supported in the theories of Kanban.

<table>
<thead>
<tr>
<th>Problem categories before the Kanban implementation</th>
<th>Evaluation of the problem categories three to five months after the Kanban implementation</th>
<th>Expected influences of Kanban</th>
<th>Outcome after three to five months</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task switching</td>
<td>Task switching is still present but more focus is put on planning and prioritizing</td>
<td>Limit work in progress which reduces task switching</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Customer collaboration</td>
<td>Customer not involved yet or no change since before</td>
<td>Customer awareness higher by among other things visualizing the work flow</td>
<td>Not supported</td>
</tr>
<tr>
<td>Lead time</td>
<td>More focus on lead times</td>
<td>Manage flow and limit work in progress which help in shortening the lead time</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Quality and incidents</td>
<td>No immediate quality effects but more discussions</td>
<td>Process policies are made explicit and limit work in progress helps increasing quality</td>
<td>Not supported</td>
</tr>
<tr>
<td>Technical debt</td>
<td>No immediate effects on technical debt</td>
<td>Process policies are made explicit and limit work in progress helps to increase quality</td>
<td>Not supported</td>
</tr>
<tr>
<td>Rework</td>
<td>No immediate effects on rework</td>
<td>Process policies are made explicit and limit work in progress helps increasing quality</td>
<td>Not supported</td>
</tr>
<tr>
<td>Lack of competence</td>
<td>No immediate effects on competence but more discussions</td>
<td>Not explicit in Kanban to increase competence</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

Table 8. The effects reached with Kanban within two months
Table 9 is displaying the problem categories the work teams had before the implementation of Kanban, the evaluation of the problem categories after using Kanban for three to five months, the expected influences of Kanban according to the theory and to what degree the outcome is supported in the theories of Kanban.

<table>
<thead>
<tr>
<th>Problem categories before the Kanban implementation</th>
<th>Evaluation of the problem categories three to five months after the Kanban implementation</th>
<th>Expected influences of Kanban</th>
<th>Outcome after three to five months</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task switching</td>
<td>Task switching is still present but even more focus is put on prioritizing</td>
<td>Limit work in progress which reduces task switching</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Customer collaboration</td>
<td>Customers understands and are aware of workload and of prioritization of work</td>
<td>Customer awareness higher by among other things visualizing the work flow</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Lead time</td>
<td>Somewhat shorter lead times by structuring work</td>
<td>Manage flow and limit work in progress which help in shortening the lead time</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Quality and incidents</td>
<td>Higher awareness and focus on quality</td>
<td>Process policies are made explicit and limit work in progress helps increasing quality</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Technical debt</td>
<td>Better cooperation in coding and reviewing code which reduces technical debt</td>
<td>Process policies are made explicit and limit work in progress helps to increase quality</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Rework</td>
<td>Higher awareness and work on quality which minimizes rework</td>
<td>Process policies are made explicit and limit work in progress helps increasing quality</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Lack of competence</td>
<td>Better cooperation and division of work items within the teams according to team members competence</td>
<td>Not explicit in Kanban to increase competence</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

Table 9. The effects reached with Kanban between three and five months
Table 10 is displaying the problem categories the work teams had before the implementation of Kanban, the evaluation of the problem categories after using Kanban for six months or more, the expected influences of Kanban according to the theory and to what degree the outcome is supported in the theories of Kanban.

<table>
<thead>
<tr>
<th>Problem categories before the Kanban implementation</th>
<th>Evaluation of the problem categories six months and more after the Kanban implementation</th>
<th>Expected influences of Kanban</th>
<th>Outcome after six months and more</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task switching</td>
<td>Focus on prioritizing and distributing work within the team but some task switching is still present</td>
<td>Limit work in progress which reduces task switching</td>
<td>Well on its way of being supported</td>
</tr>
<tr>
<td>Customer collaboration</td>
<td>Responsible and/or participating customers aware of workload and of prioritization of work</td>
<td>Customer awareness higher by among other things visualizing the work flow</td>
<td>Well on its way of being supported</td>
</tr>
<tr>
<td>Lead time</td>
<td>Somewhat shorter lead times by structuring work and focusing on workflow</td>
<td>Manage flow and limit work in progress which help in shortening the lead time</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Quality and incidents</td>
<td>Higher awareness and focus on quality</td>
<td>Process policies are made explicit and limit work in progress helps increasing quality</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Technical debt</td>
<td>Better cooperation in coding and reviewing code which reduces technical debt</td>
<td>Process policies are made explicit and limit work in progress helps to increase quality</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Rework</td>
<td>Higher awareness and work on quality which minimizes rework</td>
<td>Process policies are made explicit and limit work in progress helps increasing quality</td>
<td>Beginning to be supported</td>
</tr>
<tr>
<td>Lack of competence</td>
<td>Better cooperation and division of work items within the teams according to team members competence</td>
<td>Not explicit in Kanban to increase competence</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

Table 10. The effects reached with Kanban after six months

The result of this study can be concluded with that the outcome of the empirical study corresponds well with what is said in the theoretical frame of reference. The effects of Kanban that are described in theory are matching well with the beginning effects that we have seen in our study.
7.4 Analysis of the effects
Here is an analysis of the effects on the problem categories, additional effects and then a general analysis.

7.4.1 Effects on the problem categories

Task switching
Regarding task switching an effect of having many blocked items in a team’s work process might be that the team members are forced into task switching. There are indications that some team members feel like there is less task switching than before. The teams have started to prioritize more and they have a more explicit work process and when they start limiting WIP there will probably be stronger effects on the task switching as well.

Customer collaboration
The positive effects of collaboration with the customer are evident after some months of using Kanban. The teams that have used Kanban for up to 2 months cannot see any differences in customer collaboration yet. It seems like it is a great help to the teams that the customer is a part of the prioritization process.

The customer collaboration is easier when more and more teams are using Kanban since they have an easier time to handle internal stakeholders when they all are using the same work method. The increased feedback and communication with the customer is also one of the Lean principles. We can see that after about 3 months of using Kanban the teams are in the process of attaining this second Lean principle. When more than six months have passed participation and awareness of the customer is even higher.

Sometimes it is hard for teams when they have several customers that all think that their work items should be prioritized.

Lead time
Some of the team members say that they think they have somewhat shorter lead times, but there have not been any major changes yet. Team members say that they are working towards shortening lead times by cooperating more within the teams, but we think that any major changes in lead time will not be evident until the teams really start limiting WIP.

Quality and incidents, Technical debt and Rework
Quality and incidents, technical debt and rework are all closely related to each other and are all dependent on the quality of work. The positive effects of focusing on quality of work are also evident after three months of using Kanban, before that the teams have not seen any effects. There are some team members that after six months or more still have not seen any effects in quality.

Both the fifth and seventh principle of Lean is concerning the quality of work. Both building in integrity and seeing the whole are principles that all teams are working with now by using Kanban and focusing on quality.

It is evident that the teams are focusing more on quality than before. Many of the team members state that they discuss more within the group. The increased cooperation within the teams will after some time increase quality. The team members follow the team policies, the definitions of done and help each other with code reviews and the effect will be increased quality. In the long run common and explicit policies will help when new team members are joining the team.
Lack of competence
The lack of competence problem is not really covered by Kanban but with an increasing cooperation within the teams it is easier to get the right person on the right job, i.e. the right competence for the job. Kanban can help in creating an environment that will increase the possibility for team members to work together, do code reviews and share knowledge between each other.

7.4.2 Additional effects
Satisfaction
Regarding satisfaction, the reason why we had a question about the team member’s satisfaction with work in our survey interview was to get a general feel of what the team members felt about their new work method.

A lot of the team members like their new work method and they feel that the increased collaboration and the visualization of the work process help them a lot in their daily work. The team members also like that the stress to finish work items is now on the group rather than on the individual as it was before. Collaboration between different teams will also be easier as many of the teams have a more similar work process and have the same vocabulary.

Empowering the team is the sixth of the Lean principles. We feel that the teams have the conditions necessary according to Poppendieck’s key points to empower the team.

The Kanban implementation has given the software development teams at Sandvik positive effects on the team members own work and these effects are most evident in prioritization, visibility and work process. The effects on work process are immediate and the effects on prioritization and visibility are evident after some months of using Kanban. The Scrum using teams feel that they are getting rid of things that they did not like with Scrum, like the sprint planning.

There is someone in our study that is dissatisfied and feels that this Kanban process takes time from the work and that it does not have enough positive effects. What do one do with a person that do not like, need or feel a need for a new work method? Maybe that person has not seen enough of the effects yet. According to the other team members answers effects on for example prioritization and visibility of one’s own work takes more than three months to take effect. The change process of Kanban is evolutionary in order to make the resistance to change minimal but some people might not be ready for the change even though it is supposed to be evolutionary.

7.4.3 General analysis
Making things visible with the help of Kanban have made all the teams that we observed in the Kick start astonished of how much work in progress they really have. The power of visualizing has been apparent over and over again in our study.

We have seen that Kanban can help to remove several problematic issues that the teams have, but Kanban is not a solution for all problems. Some of the team members are unhappy that Kanban does not help them to solve their problems. It is important to remember that Kanban is not a problem solving tool, it does not solve problems but it makes them more visible to the team. The thought is that the team should evolve and increase maturity and in the end be able to solve the problems themselves with tools that are proper for each circumstance.
Regarding the time aspect we have seen that many of the effects get more evident over time, team members that are further along in the Kanban implementation process see more effects. We have seen that some of the effects of Kanban take a while to kick in, and some effects are instant. But there are some exceptions and these could have several different reasons. If a team has a lot of work in progress when they have the Kick start they will probably focus more on finishing their work than adopting the new work method. We also feel that teams that have used Scrum before have easier to adapt Kanban and already are adapted to more Lean and Agile thinking. We believe that many of the effects will be reached when the teams are further into the process of adapting Kanban, for example when the teams are staring to limit their work in progress. We have seen that some effects take months to be reached, maybe even years. Kanban is not a quick fix method even though some effects are apparent immediately.

Adapting Kanban to suit both the organizations and the team’s wants and needs is a continuing process of evaluation and re-evaluation. Kanban is used at SITS by teams that are using more traditional software development methods and by teams that are using agile software development methods, both benefiting by the adaptability of Kanban to suit the team’s needs. This adaptability of Kanban makes the process a little different for each team and this can make it hard to generalize the results and effects.

7.5 Summary of the comparison
This is our summary of the comparison of the theoretical effects and the actual effects of Kanban at Sandvik. We can see that many of the effects that we have discovered are evident in the literature. What we have not found in the literature is the time frame of when to expect effects when implementing Kanban. We have found that some effects are immediate and some effects evolve slowly over time.

This will now lead us to the conclusions and the reflections of our study.
8 Conclusions

In this part we will give our own reflections of the study. Then we will give suggestions for further research.

8.1 Our reflections

Here we will give our own reflections of this study.

8.1.1 Method and theory

Writing this report have made us compromise between different wishes; the academic part of the report needed to fit with the company specific parts of the report. Our wishes and thoughts needed to fit with our teachers’, Sandvik’s and Sogeti’s opinions. This has been a very instructive part of report writing for us. We have learned a lot by trying to balance between all the wishes and wants from the outside and our own ones.

The effects of Kanban are not so explicitly expressed in the literature. A lot of the effects that we have found in the literature are scattered among other topics and a lot of reading between the lines needed to be done to extract the effects from the literature.

We have been working two people together with this report and this has been very positive for us, we have had each other to discuss with. Especially since we have had to understand a subject that was new to the both of us it has been nice to have each other to turn to when something felt hazy or hard to understand. We have also had a lot of help from each other with writing the report in English. We are aware of that some information or nuances might have been lost in our translations.

Instead of making a survey interview with open questions like we did, we could have made a pure survey with more specific questions to evaluate. We feel that for our purpose it was better to have open questions that did not restrict the interviewees in their answering options. It is not until now towards the end of our research that we probably know what kind of questions and answering alternatives that would be the right ones to put in a pure survey. In hindsight we feel that we made the right choice to use open ended questions. The way we conducted the survey interviews with the team members are rather unusual but we feel that by conducting the interviews electronically we had the chance to interview more people and get more feedback than if we had done fewer interviews in the ordinary fashion, i.e. face to face. We focused on interviewing the team members, but we also could have interviewed some of the external customers to see if they had any comments on if they had felt any differences in the collaboration with the development teams.

We have discovered that our observations and survey interviews complemented each other. We chose to use both methods to get the opportunity to understand and cover as much as possible. When learning about Lean, Agile and Kanban, in the beginning we had some trouble both understanding it ourselves and describing it to others how these concepts are interconnected. When studying the literature different authors perceive these interconnections differently. For our own understanding we made a graph of the connections that can be seen in Figure 5. We feel that after studying this subject for two months this graph of connections that we made in the beginning is still accurate.
8.1.2 Results
We have made a qualitative research and a part of that means that we have made interpretations about what the interviewees have said. We have also made interpretations of what we have read and observed, and this subjectivity could of course have affected the results.

Since this is a qualitative study made in only one organization the ability to generalize the results might be limited. However our first and foremost priority of this study was to make a firsthand empirical study of the Kanban implementation at SITS. It is also important to point out that our assessment of the effects has taken place a relatively short time after the teams have gone through the Kick start. The result probably would differ and/or be more evident if more time had passed since Kick start since several of the teams have only just started their journey towards fully implementing Kanban.

The first set of survey interviews were distributed resulted in 10 answers. When a reminder was sent it resulted in 5 more answers. We would have wished for a higher percentage of answers, but we also feel that the later incoming results did not change the results we had gotten from the first answers from the survey interviews they rather reinforced them.

In developing the categories of description in our tables displaying the effects or actually the results, it was hard to get the category columns so clear cut that it was definite to which category each statement belonged. Both the topics and the statements we got from the interviewees sometimes became intertwined since many of the elements are affecting each other. It was not as easy to categorize statements as the phenomenographic theory said, we feel that even though we have sorted and re-sorted the categories several times some are not that clear cut. Instead of excluding statements when not finding a clear cut category for them, we instead put them in a category that mostly but not necessarily fully fitted the answer. We do not think that it has had a major affect on the result of our assessments the most important thing has been the content of the statements not necessarily how clear cut they are categorized.

In order to write this report we have studied a lot of material. The subject of Kanban in software development is fairly new which means that many of the articles, books and reports are referring back to the same books, articles and reports. Some of the authors might not have as much credibility as other authors and this has sometimes been hard for us to assess. We have had help from our instructors to discard the ones with the lowest credibility.

It was hard to make the documentation of the Kanban implementation process even though we had plenty of material to use and we had access to a lot of knowledge that our assigners helped us with. The hardest part was to write in a clear cut way in order for the reader to easily understand every step of the process. The observations were essential for us to get the first hand knowledge and experience to make the documentation accurate. We now have the utmost respect for those that produce good documentation because it is really very hard.

To conclude we think that our results are both valid and reliable. We think that our results are valid because we have gathered and analyzed the data in a systematic way which we have thoroughly described in the report. We have also used triangulation to further increase the validity. We think that our results are reliable since we had the survey interviews in writing we have had the chance to study and re-study the results several times. The analysis- and display model that we used have been reviewed by a Kanban researcher at the University of Helsinki. We have also had very good guidance from our supervisors both regarding the company specific parts and the parts concerning the reliability of sources.
8.1.3 General thoughts
We really appreciate that Sandvik has been as open as they have with their issues in the software development teams. We are certain that other companies have similar problems but might not be as open about them as Sandvik have been. Being open and honest with their problems also makes them easier to fix and we think that Sandvik is on a really good path of improving their organization for the better!

To conclude this report, one objective of this report was to assess the effects of Kanban in the software development teams at SITS. We feel that we have fulfilled this objective. The most important effects we have seen are that the teams are experiencing less stress and they have more focus on working with quality issues. The teams’ collaboration with their customers is also better now. We have also discovered that it takes time for some effects to evolve when implementing Kanban. At SITS they have Kick started the first teams a little over six months ago and we have seen that some of the effects come immediately but most effects have taken months to evolve. Probably are the major parts of the effects of Kanban still to come at SITS.

The other objective we had was to document the implementation process of Kanban in software development teams at SITS. We feel that we have fulfilled this objective as well. The documentation turned out to be extensive and was extended to include both the intended documentation of the implementation process and also a description of the process in which Kanban was chosen to be implemented at SITS.

8.2 Suggestions for further research
Our suggestions for further research are to study other companies and how they have implemented Kanban and compare it to Sandvik’s implementation process and the effects of Kanban implementations in other companies.

It could also be interesting to study the long term effects of the Kanban implementation at Sandvik and make the same study of the effects in maybe a year or two to see the development of effects.

An interesting topic could be to study how teams that are not located in the same office space are using Kanban, and study and evaluate the electronic alternatives to a physical Kanban board.

Another suggestion is to make a more extensive quantitative study of the same topics and also study why some teams see effects of Kanban quicker than other teams.
9 Glossary

**Agile** – Agile methods are software development methods that are based on iterative and incremental software development. The development is evolving in self organizing teams that often are cross functional. These methods are based on the Agile Manifesto.

**Definition of Done** – A set of principles that has to be fulfilled before a work item can be moved to the next column.

**Lean** – Lean software development is the transitioning of the Lean manufacturing and Lean Principles into software development. These were adapted from the Toyota production system.

**Lead Time** - Lead time is the time it takes a work item to pass through the workflow.

**Kaizen** – Kaizen is the approach of continually improving.

**kanban** – kanban with a small k represents the signal card used in the Kanban methodology.

**Kanban** – Kanban with a capital K represents the Kanban methodology which in a nutshell consists of three main parts; visualizing workflow, limit WIP- work in progress and measure workflow.

**Kanban board** – The Kanban board is the white board that the team uses to visualize their work process on.

**Kick start** – The Kick start is the first practical step for the team to start the Kanban implementation process. In the Kick start the team visualizes their work processes and work items, establishes policies and so on using the team’s Kanban board.

**Quick fix** – Quick fixes are to finish work with for example coding as fast as possible but not doing it properly which might cause long term negative effects on further coding and bug fixing.

**Retrospective** – The retrospectives are the occasions when the team reflects and evaluates the team’s work.

**Scrum** – Scrum is an agile systems development method that contains a set of roles and artifacts.

**SITS** – SITS is short for Sandvik IT Services. SITS is a part of Sandvik AB that supports, increases efficiency and keeps a problem free environment for Sandvik AB and their IT-users.

**SDO** – SDO is short for the Systems Development Office. SDO at Sandvik supports and helps with improving the end-to-end delivery of systems development services.

**Waste** – Waste is everything in the work process that has no value to the customer.

**WIP** – WIP is short for Work In Progress, i.e. the number of work items that is in the team’s workflow.

**Work Item** – The work item is each piece of work that every work type is divided into.

**Work Type** – The work types are groups of work items that have similar properties and mostly behave in the same way. Different kinds of work types might have for example different priorities, behaviors, focus or goals.

**Workflow** – The workflow is the streaming/flowing of the work in the work process. The flow should not be interrupted.
10 References

Literature:


On-line sources:


Company internal resources:

Oral sources:
Interview questions

1. How long ago did your team go through the Kick start?

2. What effects can you see on your own work since the implementation of Kanban?
   Please think about:
   - General thoughts about your tasks
   - Number of parallel tasks
   - Switching between tasks
   - Number of interrupted tasks
   - Prioritizing and reprioritizing of tasks

3. What effects can you see on the cooperation with the customer since the implementation of Kanban? Please think about:
   - Customer demands
   - Customer cooperation
   - Customer understanding of the Kanban work process

4. What effects can you see on the quality of your teams work since the implementation of Kanban? Please think about:
   - Quick fixes
   - Quality of code
   - Defects/Bugs

5. What effects have the implementation of Kanban had on the time aspect?
   Please think about:
   - The customer’s demands
   - Delays
   - Planning for the work items
   - Lead time for the work items

6. Have your own satisfaction with your work changed since the implementation of Kanban? How?
Kanban boards at SITS

Here are some photos of actual Kanban boards at SITS.

This is the Kanban board of a SAP (Systems, Applications and Products in Data Processing) development team. The picture was taken one week after the Kick start.

This Kanban board belongs to a team working with .NET development. The team was using Scrum before being Kick started. The picture was taken a couple of weeks after the Kick start.
This is the Kanban board of a team working with Main Frame development. This team also used to use Scrum before using Kanban. This picture was taken a couple of months after the Kick start. On this Kanban board it is noticeable that there are WIP limits, though they are not enforced and must be re-evaluated.

The picture shows the Kanban board of a team working with Main Frame development. This team was the first team to try Kanban at SITS during the fall 2010.
This picture shows the Kanban board used by the Competence Managers at SITS (line managers). The team is in full transition, focusing on finishing the work in process and refraining to start new work. This picture was taken a couple of weeks after the Kick start.

This picture shows the Kanban board used by the SDO.