Benefiting of improvements in Scrum project regarding web application
- A Case study at T-Systems Sweden

Lior Azouri and Marcus Olsson
This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 2x20 weeks of full time studies.

Contact Information:
Author(s):
Lior Azouri, lior.azouri@gmail.com
Marcus Olsson, marcus.h.olsson@gmail.com

External advisor
Jan Karström
T-Systems Sweden AB
Address: Jörgen Kocksgatan 4
Phone: +4640-6311415

University advisor(s):
Miroslaw Staron
School of Engineering, Department of Systems and Software Engineering

School of Engineering
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Internet : www.bth.se/tek
Phone : +46 457 38 50 00
Fax : + 46 457 271 25
ABSTRACT

This thesis describes Scrum and with a feasibility study tests if Scrum can be applied in a project with two developers at T-systems Sweden. Key issues regarding the development process are pointed out from a baseline project. The influence of the issues in the project is later tested in the new project at the same organization. New issues were identified and studied and corrected. Interviews and literature study are used to gather information of what issues do occur and may occur. The results and analysis lead to a conclusion on what changes that should be done and what strength and weaknesses Scrum has when implemented at T-systems Sweden.

Keywords: Scrum, web application, software development method, software engineering
# CONTENTS

## ABSTRACT

CONTENTS ................................................................................................................................. II

1  INTRODUCTION ..................................................................................................................... 1

   1.1 BACKGROUND .................................................................................................................. 1

   1.2 RELATED WORK .............................................................................................................. 2

   1.3 RESEARCH METHODOLOGY ......................................................................................... 2

   1.4 THREATS TO VALIDITY ............................................................................................... 2

      1.4.1 Conclusion validity ................................................................................................. 2

      1.4.2 Internal validity ...................................................................................................... 2

      1.4.3 Construct validity .................................................................................................. 2

      1.4.4 External validity .................................................................................................. 3

   1.5 TARGET GROUP .............................................................................................................. 3

   1.6 DEFINITION OF AGILE ............................................................................................... 3

      1.6.1 Agile manifesto ..................................................................................................... 3

   1.7 DEFINITION OF METHOD, METHODOLOGY AND PROCESS ........................................ 5

   1.8 DEFINING SUCCESSFUL PROJECTS ............................................................................ 5

   1.9 RESEARCH QUESTIONS ............................................................................................... 6

2  DEFINING SCRUM .................................................................................................................. 7

   2.1 BACKGROUND .............................................................................................................. 7

      2.1.1 Scrum process ........................................................................................................ 7

      2.1.2 Undefined elements in Scrum .............................................................................. 8

      2.1.3 Philosophy ........................................................................................................... 8

   2.2 ROLES ............................................................................................................................ 9

      2.2.1 Scrum master ....................................................................................................... 9

      2.2.2 Scrum team ......................................................................................................... 9

      2.2.3 Product owner .................................................................................................... 10

   2.3 COMMUNICATION AND INFORMATION ..................................................................... 10

      2.3.1 Daily Scrum meeting ......................................................................................... 10

      2.3.2 Sprint planning meeting .................................................................................... 11

      2.3.3 Sprint – 30 days iteration .................................................................................. 11

      2.3.4 Review meeting ................................................................................................. 12

      2.3.5 Sprint retrospective meeting ............................................................................ 13

   2.4 DOCUMENTATION ....................................................................................................... 14

      2.4.1 Product Backlog ................................................................................................. 14

      2.4.2 Sprint Backlog ................................................................................................... 14

      2.4.3 Burndown chart ............................................................................................... 14

      2.4.4 Sprint Goal ......................................................................................................... 15

3  DEVELOPMENT AT T-SYSTEMS ........................................................................................ 16

   3.1 BASELINE PROJECT ................................................................................................... 16

      3.1.1 Scrum and Extreme Project Management .......................................................... 17

      3.1.2 Project methodology in the baseline project ....................................................... 19

      3.1.3 Roles .................................................................................................................. 19

   3.2 NEW PROJECT ............................................................................................................. 20

4  RESEARCH METHOD .......................................................................................................... 21

   4.1 CASE STUDY ................................................................................................................. 22

      4.1.1 Evaluation of the baseline project .................................................................... 22

   4.2 MEASUREMENTS ......................................................................................................... 23

5  ANALYSIS OF CASE STUDY ........................................................................................... 25

   5.1 METRICS ....................................................................................................................... 25

   5.2 CASE STUDY ................................................................................................................. 27
5.2.1 Results from questionnaire ................................................................. 27
5.2.2 Results from interviews ........................................................................ 29

6 FEASIBILITY STUDY ................................................................................. 32
  6.1 SPRINT 1 .............................................................................................. 32
  6.2 SPRINT 2 .............................................................................................. 33
  6.3 SPRINT 3 .............................................................................................. 34
  6.4 STRENGTHS IN SCRUM ......................................................................... 35
  6.5 WEAKNESSES IN SCRUM ....................................................................... 35

7 CONCLUSIONS .......................................................................................... 36

8 FUTURE STUDIES ...................................................................................... 37
  8.1 BIGGER TEAMS .................................................................................... 37
  8.2 CUSTOMER AND STAKEHOLDERS ...................................................... 37
  8.3 DIFFERENT KINDS OF PRODUCTS ....................................................... 37

9 GLOSSARY .................................................................................................. 38

10 REFERENCES ............................................................................................ 39

11 APPENDIX A. QUESTIONNAIRE .............................................................. 41

12 APPENDIX B. MEASUREMENT DATA FROM SOURCEMONITOR .......... 49

13 APPENDIX C. QUESTIONS FOR THE INTERVIEWS ................................. 54
1 INTRODUCTION

1.1 Background

When a software company is small the projects in that company are small as well. When projects are small teams are small as well but when the projects expand, the teams expand. When teams expand new problems may occur. The projects have to have a structured way of working. T-systems is a company on expansion and has in the past year grown to be so large that the projects need a structured way of working.

The objective in this thesis is to analyse the development process in a web application project that uses parts of Scrum. The second part is to find out how Scrum can fit T-systems organisational structure.

Scrum is a development process that can be used to produce any product or manage any work [3]. In the research a case study has been made and key issues from a baseline project have been tested in a new web application project. The research is made to test how Scrum fits projects on T-systems Sweden. A typical project at T-systems has two to three developers.

A formal Methodology contributes with six percent to a projects success according to Standish groups report “Extreme chaos” from 2001 [4]. Even if six percent can be seen as a poor factor, it still is a factor. This thesis intends to inform the reader of Scrum and how it can help T-systems projects as a formal methodology. This thesis can also help project managers to understand how Scrum relates to the work effectiveness in the project.

The first part of this report explains how the research is structured. The second part clarifies Scrum and how the baseline project works. The last part consists of result and a conclusion of the work.

1.2 Related work

In order to understand and get an objective view of Scrum as a development process, published papers and books have been read. The contents of both the papers and the books have been Scrum and other Agile processes related. Guidelines regarding interesting literature have been gathered from Scrum’s official newsgroup [15].

There are not many books or rapport published on Scrum that objectively describes how Scrum is working and none of the rapports that have been found were criticising Scrum. The reports describe how Scrum have been implemented and adopted by a company and how it leads to a better way to carry out projects. If the reason for that is that only success stories are published, no one has failed with Scrum or if they that have failed have not publics in any articles.

L. Rising and N. S. Janoff in the article “The scrum software development process for small teams” from 2000 the author is writing about how Scrum works in small teams [20]. Other articles that describes how Scrum have been adopted is “Primavera Gets Agile” by B. Schatz and I. Abdelshafi from 2005 [21] and “Improving the Interface Between Business and Product Development Using Agile Practices and the Cycles of Control Framework” [22].

No related work has been found of any consultant company are using Scrum.
1.3 Research methodology

Both a quantitative and qualitative method has been used in order to gather information for the research. The quantitative method includes interviews and a questionnaire with developers. The qualitative part includes testing Scrum as a development process in a new project. A company called Skånetrafiken was initially chosen as the customer for the new project. When the pre-study was completed Skånetrafiken withdrew from the new project and the internal advisor at T-systems acted as customer.

1.4 Threats to validity

1.4.1 Conclusion validity

“Conclusion validity is the degree to which conclusions we reach about relationships in our data are reasonable.”[16] The conclusion validities are:

**Threat:** Low statistical power due to low sample size

**Minimize threat:** This threat can not be removed. There exist only four developers in the project that is to be studied.

**Threat:** Low statistical power due to different size between the two projects

**Minimized threat:** Try to compare on common ground.

**Threat:** Hard to compare the two projects

**Minimize threat:** Instead of comparing, the studied project was seen as a baseline project.

1.4.2 Internal validity

“Internal Validity is the approximate truth about inferences regarding cause-effect or causal relationships” [17]. Does the study change what was observed?

The internal validity in this research is:

**Threat:** The observer of the baseline project became the observed in the new project. This means that the study looses its objectiveness when the new project started.

**Minimize threat:** By this the objectiveness could deteriorate. Since the feasibility study had to be made this threat could not be removed.

1.4.3 Construct validity

“Construct validity involves generalizing from your program or measures to the concept of your program or measures” [18], thus the different between how the study was thought to be set up and how it was really constructed.

**Threat:** The new project is less prioritized than the baseline project.

**Minimize threat:** To be able to minimize this threat greater planning and more persuading of the new project is needed.

**Threat:** When the study was initiated it was based on that Scrum was fully used in the baseline project.

**Minimize threat:** The study changed direction from using Scum to using parts of Scrum.

**Threat:** Customer withdrew from the new project.

**Minimize threat:** Since the Customer withdrew from the new project the part of Scrum the contained customer relations could not be appropriately studied.
1.4.4 External validity

“External validity is the degree to which the conclusions in your study would hold for other persons in other places and at other times” [19].

The external validity in this research is:

**Threat:** The developers and project managers in the baseline project are not interesting for the reader of the results.

**Minimize threat:** The objects are chosen to be system developers with bachelor’s degree.

**Threat:** The new project is not a usual Scrum project. Since the new project only consist of two developers it clashes with Scrum

**Minimize threat:** According to L. Rising and N. S. Janoff [20] Scrum can be applied to small project as well.

1.5 Target group

When projects tend to get large enough that ordinary conversation between developers is not enough, successful project structure has to play a bigger role. F Brooks suggests that the effort of intercommunication is increasing by n*(n-1)/2 [8].

This report is directed to projects managers in software projects who want to structure their projects with the help of Scrum, organisations that want to adopt Scrum and developers that are interested in the positive and negative aspects of Scrum. The report is also interesting for research of Scrum as an agile development process.

1.6 Definition of Agile

An agile software method is a way of describing an agile approach to projects.

Agile is described as “Having the faculty of quick motion, nimble, active, ready” in Oxford’s English Dictionary. A group of people set up a meeting in early 2001 to define a general process that contains quick respond to change. This group was called the Agile Alliance and the process they developed was The Manifesto. The Manifesto [1] includes the principles of Agile software development. These principles are briefly described at the next chapter. [1]

The Agile processes are [11]:

- Scrum
- Extreme Programming
- Chrystal Family
- Feature Driven Development
- Adaptive Software Development
- Dynamic Systems Development Method

1.6.1 Agile manifesto

In order to get a better view of what Agile is in the meaning of development processes this chapter will quote the Agile manifesto [1]. Below each quote (in italic) there is a description to the quote.

- **Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**
  Instead of deliver a final product at the end of the project, working functionality is delivered to the customer throughout the project. The customer has the chance to see how the project is progressing, not in
measures but in working functionality. Even Frederick P. Brooks [8] applies that it is impossible for the client to completely, precisely and correctly specify what exact requirements to set to a modern software product. That is why a continuous delivery is satisfying for the customer.

- **Welcome changing requirements even late in development. Agile processes harness change for the customer’s competitive advantage.**

Instead of having one static plan the requirements are changed throughout the project, even in the end. It is the customer who decides how the progress shall be made.

- **Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time scale.**

The development team delivers working software frequently with a smaller timescale, instead of one product at the end of the project.

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

Developers, customers and developers have to have frequent interaction in order to maintain continuously guided projects.

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

The individuals are placed in centre of the project. They want to and have to decide what environment they want.

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

Communication via Documentation should not be used. By talking to each other directly misunderstanding is avoided and it takes less time.

- **Working software is the primary measure of progress.**

Working software is the only progress that is real. Phases and documentation is not a way of telling how much of the features that are working.

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

By having developers work overtime, they work themselves out. To have a constant pace both in the beginning and the end the developers don’t wear themselves out.

- **Continuous attention to technical excellence and good design enhances agility.**

Only high quality code is allowed.

- **Simplicity--the art of maximizing the amount of work not done--is essential.**

This is to have a minimised scope i.e. not looking at and solving tomorrow’s problems but today’s. The Standish group also referrers to a minimized scope as 10% of the successful factors in a project [4].
• The best architectures, requirements, and designs emerge from self-organizing teams. Responsibilities and activities should not be given to people inside or outside the team but should be chosen by the team members themselves.

• At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly. The organisation, rules, conventions, relationships of the team is continuously changing, everything to make the team as effective as possible.

1.7 Definition of Method, Methodology and Process

In Merriam-Webster dictionary the word method was explained as “systematic procedure”, similar to a technique. The word methodology was explained as “A series of related methods or techniques”. This makes a methodology an integration of many methods.

In the same dictionary the word process is defined as “a natural phenomenon by gradual changes that lead towards a particular result”. A process is thus how things develop over time toward a result.

In Ian Sommereville’s Software Engineering [6] the author describes a software process as “a set of activities and associated results which lead to the production of a software product”. All processes consist of these fundamental phases:

- Software specification
- Software design and implementation
- Software validation
- Software evolution

Scrum is mostly seen as a process but can also be arrange to work as a methodology with different phases. In the report from Cockburn [12] the author defines process as “how activities fit together over time, often with pre- and post conditions for the activities”.

1.8 Defining successful projects

When starting this research the expression successful project was discussed. A research was made and an organisation called “The Standish group” that studied project for over ten years was used to find out different definitions [4]. The Standish group has defined an expression for what a successful project is, but also defining challenged and failed projects. Furthermore they have defined factors for projects to be successful. In this chapter these three definitions are quoted. A description of what role the Formal methodology has in a project is also explained. They define the projects as [4]:

Successful: The project is completed on time and on budget, with all features and functions originally specified.

Challenged: The project is completed and operational, but over-budget, over the time estimate, and with fewer features and functions than initially specified.

Failed: The project is cancelled before completion or never implemented.

By having a formal methodology lessons can be made to assist the work in future projects. The report does also show that in the successful projects, 46 percents had a formal methodology while in the failed 30 percent had one. This means that chance of success increases with 16 percent in projects that have formal methodology.

The Standish group also publishes that the ingredient in a successful project completion are good stakeholders, an iterative development process, project
management tools and key roles. To our best knowledge no publications were made in figures on these ingredients.

1.9 Research questions

The purpose of the study is to suggest improvements in T-systems’ development process in a project regarding a web application. This will be achieved through analysing Scrum to know how the process works. An objective study is also made to find out which practices could be developed into impediments in a web application project. In the thesis a study has been made to find out how Scrum fits into T-systems Sweden’s organisation. A feasibility study was performed in order to verify whether the identified practices could increase the efficiency and work at T-Systems.

Main question
What impediments occur when using Scrum at T-Systems Sweden?
To be able to achieve this three iterations have been completed and after each iteration impediments have been analysed, reviewed and solved before the next iteration.

Sub questions

- What strengths and weaknesses exist using Scrum in complementing modules?
- What are the different opinions of the roles regarding development processes?

Scrum’s process consists of both positive and negative aspects. In this report both these aspects will be clearer for the reader. Different roles in the organisation have different opinions on the development process. Interviews have been made to find out what different roles feel and these answers have been analysed.
2 DEFINING SCRUM

2.1 Background

Scrum is a “process for developing any kind of product” [5]. This makes Scrum a development process. Its values are based on values from Agile development methods and thus also included in the Agile Alliance.

Instead of planning to build the right product from the beginning, the phases are worked through the entire project. This is reached by close contact with the customer. At the end of each iteration an increment of the product is made and reviewed by the stakeholders. By doing this, the customer can get a feeling on how the product is evolving and can make changes and corrections during the entire project. Tests and documentation is completed during each Sprint and this leads to better quality of the final product [3]. In Scrum the Project manager is called Scrum master, the team is called Scrum team and the responsible for buying the product (usually customer) is called Product owner. The requirements list in Scrum is called the Product Backlog and contains prioritised requirements for the whole project. An iteration in Scrum is called a Sprint and lasts 30 calendar days. The document that contains the activities and requirements for a sprint is called Sprint backlog and is prioritized as well. At the end of the Sprint a Sprint review meeting is held to show all stakeholders what have been produced in the Sprint. The last meeting in the Sprint is called Sprint retrospective meeting and is held to help the Scrum team to be more effective in the next Sprint. The next chapter explains how all these things are glued together.

2.1.1 Scrum process

To explain the process better a SPEM diagram in figure 2.1 will show all steps defined in Scrum.

The process starts with the election of a Product Owner. Usually the management have a set of requirements that are written down into a Product backlog by the new elected Product owner. The management puts together a Scrum team with a Scrum master. From this point the Scrum master is responsible to set up all the meetings defined in Scrum. The first meeting in a Sprint is the Sprint Planning Meeting. In the first part of this meeting the Scrum team, Product owner and other stakeholders meet to discuss which requirements from the Product backlog should be worked on in the next Sprint. In the next part of this meeting a prioritised Sprint backlog is produced by the Scrum team. The Scrum team then decides how to break down each requirement into smaller activities.

During the Sprint all activities are completed and this produces an increment of the product. At the end of the Sprint a Sprint Review Meeting is held to show all stakeholders what have been produced in the Sprint. The last meeting in the Sprint is the Sprint Retrospective Meeting. This meeting is held for the developers to share their view on what went well and what could be improved the next Sprint.
2.1.2 Undefined elements in Scrum

Scrum does not represent a complete model for developing products. It only describes how the process of development should be and not how different phases in the development should be performed. Scrum does not point out how the definition phase of a project should be performed. The first step in Scrum is thus the planning meeting between the developers and the customer.

2.1.3 Philosophy

By only following the rules and roles setup by Scrum the project is not assured to reach success. It is also important for the group to understand what Scrum and Agile consists of. The group and the Scrum master have to change their way of thinking to achieve success. One of Scrum’s essentials is that the group is self-organised. In Scrum this means that no one outside the development group has the authority to tell the developers how to solve a certain problem. This sets new requirements on the Scrum master. The Scrum master’s role in a Scrum team is to act as a firewall to rest of the world. An example is that the management wants to interfere with the Scrum
team. Then the Scrum master has to act as a protector so that the Scrum team can work without interference. The role of the Scrum master is explained later in this chapter.

To be able to work as Scrum proposes, the Scrum team together has to have good knowledge in analysis design, implementation, testing and documentation. This leads to self-organisation and cross functionality.

Iterative and incremental are concepts that are in focus in Agile development. In Scrum an iteration is called a Sprint and lasts 30 calendar days. At the end of each Sprint an increment should deliver working functionality. The team and the Scrum master have to start thinking in Sprint and have to focus on functionality that the particular Sprint is going to implement, not the whole project. This means that the analysis, design, implementation, testing and documentation in a Sprint should only cover the functionality being implemented during a particular Sprint.

Scrum is an empirical method. That means that it is not static but has to be changed all the time, this in order to fit each company and project as well as possible.

### 2.2 Roles

Scrum only has three defined roles. Those are Product owner, Scrum master and Scrum team.

#### 2.2.1 Scrum master

Instead of having a project manager in Scrum, a Scrum master is appointed. The reason for changing the name from project manager to Scrum master is that the role for the Scrum master differs from the role of the project manager. In Scrum the Scrum master is responsible for helping the Scrum team with tasks that concerns other things than development. This could for example be having meetings with the customer, top management and other stakeholders. The Scrum master can help the team with important decisions that the team can not decide by itself and is also acting as a firewall between the team and the rest of the world. It is important for the Scrum master to understand that he or she should not control the team but only guide it. It is up to the Scrum master to inform the stakeholders and the team of the rules and values of Scrum. Another important responsibility is to help the product owner with the product backlog. It is the Scrum master’s responsibility to setup and lead all the meetings.

To help the Scrum Team work as smoothly as possible the Scrum master has to handle all the problems that occurs regarding the project. These problems can appear inside the team, against the management, customer or lack of knowledge in the team.

#### 2.2.2 Scrum team

The Scrum team is self organised and is cross functional. Self organisation means that the team by itself decides who will do what. Everyone in the Scrum team has the same responsibility and authority.

Cross functionality means that every team member beholds a broad knowledge. There is not any model of the structure and what knowledge should be contributed to the team. The size and complexity of the project decides this.

This does not mean that everyone has to be expert in every field.

The size of the Scrum team should be between five and nine persons. The reason for this is that if it is less than five developers there is a risk that the team will not have enough knowledge width. If the team consists of more than 9 developers the team looses the easy communication between each other.
A solution to bigger teams is to split them up into smaller teams so that each team is no less than 5 and no more than 9 people. If there is more than one team working in a project the Scrum masters meet after the Daily Scrum to discuss. This meeting is called Scrum of Scrums.

If the team feels as if it was their project and their responsibility they will be more committed and motivated. By not having a predefined structure in the group, the group will be more flexible and can easily adapt to new conditions. It is not the management, Scrum master or the Customer that know best how the work should be completed, it is the Scrum team. That is why the responsibility and confidence is put on the Scrum team.

2.2.3 Product owner

The product owner is the person that is responsible for the overall requirements, return of investment (ROI), the goal with the project and the project as a whole. The product owner is also responsible to update the requirements in a list called product backlog and the prioritizing of these requirements are also made by the product owner. The product owner does not have to do all this by himself but can take help from other advisors.

It is important that all the involved in the project have one person and only one person that is responsible for having an overall view and responsibility for the project as a whole.

2.3 Communication and information

2.3.1 Daily Scrum meeting

A Daily Scrum meeting is a 15 minutes long morning meeting where the Scrum team (everyone) meets to summarize what has been done and what is going to be done in the next day. All problems that may occur are also brought up in this meeting. The meeting is led by the Scrum master. No kind of solutions should be discussed during this meeting. The questions that everyone in the Scrum team should answer are:

- What have you done since the last Daily Scrum regarding this project?
- What will you do between now and the next daily Scrum meeting regarding this project?
- What impedes you from performing your work as effectively as possible?

Everyone except the person that answers the questions must be quiet. Everything that does not answer these three questions must be left out of the meeting. If anyone wants to discuss something else they should meet after this meeting to discuss. Other people involved in the project can also attend but have to be quiet during the entire meeting.

This meeting is held to help the Scrum team to synchronize their work and for them to know what everyone is doing. These meetings are also held to inform others in the team of obstacles so that everyone can help each other during the day. The Scrum meeting removes parts of the intercommunication problem where the developers have to communicate with n*(n-1)/2. [8]
### 2.3.2 Sprint planning meeting

This meeting consists of two four-hour planning meetings and is held before the beginning of each Sprint. The first of these two meetings the Scrum master, Scrum team, management, possible stakeholders and the product owner meets to discuss what should be completed in the next Sprint. The most prioritised functionalities in the product backlog are selected to be completed in the next Sprint. The Goal for the Sprint is also decided (explained later).

![Sprint Planning Meeting Diagram](image)

**Figure 2.2 Sprint planning meeting**

The second part of the planning meeting is for the Scrum master and the team and is held to break down the functionalities that are chosen to smaller activities into a Sprint Backlog. The activities should take roughly between four and 16 hours. Sometimes activities may not be possible to break down in the beginning and could therefore be broken down later in the Sprint. An overall design should be made in this meeting.

The first meeting is held so that everyone involved in the project together can decide what should be made in next Sprint. The second meeting is held so that the team can decide how the work should be completed.

### 2.3.3 Sprint – 30 days iteration

This Sprint is 30 calendar days long and during this time the product is developed. The iterations keep going until the stakeholders decide that the product is complete, or the funds have run out. At the end of every Sprint a working product should be presented. The end date for the Sprint can not be changed. If there is too much work in the Sprint Backlog and everything can not be completed in time the functionality that is not completed is moved back to the product backlog. If there is too little work, functionality is added to the Sprint backlog from the product backlog with help from the product owner. The functionality in the Sprint backlog can not be changed by anyone outside the team. The Goal with the Sprint is that the Sprint Goal is met. Management, the product owner, the team and the Scrum master have the authority to end a Sprint if they feel like the Sprint is no longer useful or possible to full fill.

When the Sprint ends a product should be possible to deliver i.e. test and documentation should be completed.
The reason for this time period is that an iteration of 30 days is enough to build something useful but not too much time to lose if the direction is wrong.

Figure 2.3 Sprint

2.3.4 Review meeting

At the end of each Sprint a four hour long Sprint review meeting is held for the Product owner and stakeholders by the Scrum team to show what has been completed.
in the last Sprint. The team should only demonstrate completed functionality in this meeting and Power Point should not be used. The preparation should not take more than one hour. Strength and weaknesses with the solutions should also be presented in this meeting. The planned Sprint Goal and the demonstrations are compared to see if the team has completed what they set up to do.

Figure 2.4 Sprint review meeting

This meeting is held to show all involved what has been completed in the last Sprint and to decide what should be made in the next. The customer also has a chance to change direction on the future development of the product. This meeting helps to motivate all parts involved in the project.

2.3.5 Sprint retrospective meeting

After the Sprint Review meeting a three hour long retrospective meeting is held. The goal with this meeting is to take the experiences from the last Sprint, review them, change what should be changed, and take that experience to the next Sprint. This meeting is held for the Scrum master, Scrum team and optionally the Product owner. Everyone should answer two questions and these are:

- What went well during the last Sprint?
- What could be improved in the next Sprint?
The Scrum master’s role is not to give answers but to write down the conclusions in the meeting. The issues are prioritised and discussed by the team. Solutions can be put in the Sprint backlog as non-functionality requirements.

This meeting is held to improve and to make the next Sprint more efficient. It is also held to remove impediments in the next Sprint.

### 2.4 Documentation

#### 2.4.1 Product Backlog

The requirements in Scrum are written down in a prioritised list called a product backlog. This list should be constantly updated by the Product owner. The requirements and changes can be suggested by anyone involved in the project but can only be changed and prioritised by the product owner. There is a possibility that all functionalities on the list are not needed or wanted by the product owner and will not be completed at the end of the project.

The product backlog is also made to have an overall view of the functionalities. To help all the involved to have a document that describes the functionality of the whole project.

#### 2.4.2 Sprint Backlog

The Sprint backlog is a prioritised list of activities that are planned to be completed in the Sprint. It is only the Scrum team that has authority to make changes in the Sprint Backlog. The Sprint backlog is constantly updated during the Sprint. A Sprint backlog should consist of activities, responsible for that activity and time to completion for each activity. It is up to each developer to update time to completion at the end of every work day. That is why it is important that every developer has access to this backlog. The Scrum master is responsible for the Sprint backlog.

It is important to have a constantly updated list that describes what should be completed and how the Sprint is planned.

#### 2.4.3 Burndown chart

Two kinds of Burndown charts exist in Scrum. The Product Burndown chart shows how many days of work that remains and how much is completed during earlier Sprints. This is shown in Figure 2.
The Sprint Burndown chart shows how many hours remains until all requirements are completed during the Sprint as in Figure 2.7. As seen on this chart a trend line with the equation $y=kx+m$ can be added. This is made to get a better view of the work remaining. The trend line should reach the bottom at the end of the Sprint. If the line does not reach the bottom right corner this means that the activities will not be competed in time and the Sprint could fail.

2.4.4 Sprint Goal

During the Sprint planning meeting the Goal with the Sprint shall be written down. The goals with the Sprint should be easy to read and easy to find by anyone in the project.

As sport teams need a goal to succeed even a development team need one. The goals form the team and direct it in the right direction [13].
3 DEVELOPMENT AT T-SYSTEMS

T-systems is a company that both provides consultancy services and conducts “in house” projects.

An average software development project at T-systems consists of a set of different roles that are active during different phases. In the first phase the business management deals with the definition and vision of the project (Figure 3.2). This phase deals with defining the requirements. In the second phase the development team and the project manager is chosen. The development usually consists of developers of code (both GUI and database), a usability advisor and in certain projects an art director. In this report we call the usability advisor and art director experts. These experts are usually involved in several projects simultaneously and are often involved in the beginning of the development phase.

A project at T-systems usually consists of 2-3 developers and they usually develop their software during a time of 2-3 months. Occasionally developers can be moved between projects due to prioritization by the top management.

3.1 Baseline project

The baseline project was a Customer Relationship Management project called Boomerang CRM phase 2. The application manages claims from customers. Boomerang CRM is divided into different parts.

The first part is the Boomerang CRM framework. It is in this framework the tested program is developed. The framework consists of predefined controls in ASP and a certain standard that all projects in boomerang follow.

The second part is Boomerang CRM for credit cards. The second and third Sprint in this part is what is referred as the baseline project in this research. The second sprint consisted of development and the third Sprint of testing. Boomerang CRM is a web application and is divided into four different layers. The first layer is the web UI layer and is developed in Visual Studio .NET. It uses ASP.NET with C# as background code. The second layer of the baseline project is the facade layer and is also developed in Visual Studio.Net with C#: The third layer in the baseline project is the business layer and is developed in Visual Studio 6 with Visual Basic. This layer interacts with the database layer in SQL Server.

The project manager in the baseline project decided to use parts from Scrum and Extreme project management [5] to run the project. The parts in the baseline project that have Scrum approach are:

- Sprint and Product backlogs
- Sprints (not always exactly 30 days)
- Daily Scrum meetings
- Iterative development
- Burndown charts

To get a better view of the methodology used in the baseline project it is necessary to understand what Extreme Project Management is and the differences and similarity between it and Scrum. In chapter 3.1.1 this is explained and in chapter 3.1.2 and 3.1.3 a more complete view of the project methodology and roles that were used in the baseline project are explained.
3.1.1 Scrum and Extreme Project Management

The project methodology T-Systems Sweden used in the baseline project was a combination between Extreme project management and Scrum. Scrum and Extreme project management (XPM) are very similar on many points even if they at a first glance seem to be very different. The XPM methodology is more defined, has more rules and is more directed towards the project manager than the developers unlike Scrum. This is the biggest difference between Scrum and XPM [7].

The structure and the mindset of the development teams are almost the same. They both consist of cross-functional teams that manages them self and turns to the project manager only when they need help with problems and decisions. The project manager’s role is to manage the business side of the project and see to that the value of the methodology is followed during the development.

The iteration phase is similar in both models. The teams should be constant and if the requirements are changed the iteration is abandon and a new one is started. One difference is that Scrum advocate a fix time of 30 calendar days for a iteration when XPM:s is between one to six weeks.

XPM is directed more towards the project manager and business part of a project than to the developer and thus contains a set of Business questions to ensure that the customer get the right product during the project.

- Question 1: “Who needs what and why?”
- Question 2: “What will it take to do it?”
- Question 3: “Can we get what it takes?”
- Question 4: “Is it worth it?”

The project model XPM is using a model called the flexible project model (FPM). This has the four business questions as a base (Figure 3.1). FPM is an iterative development model with five cycles; Visionate, Speculate, Innovate, Reevaluate and Disseminate that contains checklists and goals.
During the Visionate cycle the goal is to get the vision from the customer and mediate this vision to the development team. The Visionate cycle contains a set of interviews with the customer to get an answer to the first business question “Who needs what and why?” At the end of the Visionate cycle an estimate of what is needed to complete the project is made and the risks that can occur are considered. The Visionate cycle ends with a meeting with the customer to get a go or not and a development team is picked for the project.

The Speculate cycle consists of getting an understanding of what has to be completed during development, how it is going to be done and if it is worth to do. This cycle starts with a planning meeting involving the project manager, developers and other parts that have information necessary to the project. The outcome of the planning meeting should be a plan of how the product is going to be developed and a schedule of the project. The cycle continues with choosing how the project is going to be managed and what tools and templates are going to be used. The ROI is also estimated and establishment contact with external partner is made if needed. Before moving into the development cycle the second business question should be answered and question number three and four is addressed.

The first development cycle is called innovate. During this phase an increment of the product is created. This cycle is time boxed to be one to six weeks and starts with a meeting where developers meet to discuss how the work shall be completed. At the end of the Innovate cycle recommendations for future work are being done.

During Revaluate the project is reevaluated if it needs further development, should be released or cancelled. The Revaluate cycle is taking place every three to four weeks in short-duration projects that spans over three to six months in length. For longer projects it takes place every two or three month.
During the Disseminate cycle the product is released and the project is reviewed for future projects [7].

### 3.1.2 Project methodology in the baseline project

As mentioned earlier the project methodology in the baseline project is a mixture of Scrum and XPM. Figure 3.5 shows a SPEM diagram over the process used in the baseline project.

![Diagram showing project methodology](image)

**Figure 3.3 Baselines project model**

As can be seen in figure 3.3 the baselines project model have more and specialized phases for the different states in the development than Scrum has. In the definition sprint most of the definition is done but some is done in the development sprint. The test sprint is between two to three weeks and ends with the customer doing an acceptance test. Rollout sprint is when the product is released to the customer.

Only the innovation and test sprint were thoroughly studied.

### 3.1.3 Roles

The roles in the base line project are project manager, technical advisor, usability advisor and art director.

The project manager’s role is to be the link of communication between the sponsor and the customer, plan the project and to make sure that the project has all the
resources it needs. It is the project manager that has the final saying in how things shall be done an implemented.

Technical advisor has the responsibility for the technical parts of the project. The technical advisor hands out the different tasks to the developers and has the responsibility for architecture of the product.

Developers in the baseline project are more or less specialised on one part of the development. For example the person that implements the UI layer has little knowledge of how the business and data access layer is working and the person that develops the business layer has little knowledge of the UI layer.

To assure that the solution is going to be user friendly a usability advisor is used. The usability advisor have knowledge of how different users interacts with a certain type of program and if needed study how the users are interacting with the program. Then recommendations for how the UI and the work flow of the program are made to the project manager and developer.

The art director’s role is to develop icons and design if needed.

### 3.2 New project

In order to test and evaluate Scrum a project called Lost and Found was carried out. This project began with a definition and vision phase. This phase included interviews with the customer. After the interviews and revising with business management at T-systems a definition document was written. This document consisted of what could be developed in the project.

The requirements were written down and prioritized in a product backlog. Three Sprints were carried out to test Scrum, where each Sprint was three and a half weeks (calendar days) long. All practices in Scrum were tested but also key issues from the baseline project regarding development processes were tested in the project. Lost and Found was thought to be a module in Boomerang CRM and thus had to follow all standards and framework that CRM followed. The developers in Lost and Found were Lior and Marcus. Besides the developers, the new project had access to a user experience advisor, Art designer and a technical advisor. The customer (Product owner) was an employee at T-systems.
4 RESEARCH METHOD

Figure 4 describes the phases of the research over time. The process started with planning the thesis (A).

The first three things that had to be completed were a pre-study of the new project (C), an evaluation of the baseline project (D) and a study of development processes (B). The pre-study of the new project consisted of interviews with the customer and defining requirements in the project. The evaluation of the baseline project consisted of having interviews with developers and project managers to be able to set up key issues regarding the development process in the baseline project (E).

Research of the development processes (B) was a phase that consisted of studying literature and published papers regarding Scrum. This study lead to define a way of working in the new project and writing the report. The study of development processes together with the evaluation gave a picture on how the baseline project’s development process was build up and how it related to Scrum. When the study of the baseline project and of the development processes was completed, the pre-study phase of Boomerang Lost and Found and the development phase of Boomerang Lost and Found was completed, the development began (H). The development phase in Boomerang Lost and Found was carried out to be able to test if the same problems occurred if a development model was used and understood by the developers. This development phase consisted of three Sprints where each iteration resulted in a set of key issues in Boomerang Lost and Found. These key issues gave the project an opportunity to review the way of working (I).

![Figure 4.1 Research process](image-url)
4.1 Case study

The case study is divided into three parts, questionnaires, interviews and a feasibility study. The questionnaires and the interviews are used to get both a quantitative and a qualitative research. A feasibility study is also made to test whether it is possible to use Scrum as a project methodology at T-systems and if impediments occur how to remove these impediments. The reason for not using the same developers in the new project as in the baseline project was because the developers in the baseline project were involved in another project.

The goal of the interviews is to get the opinion of the different roles regarding agile development and an understanding of how the organisation is working.

4.1.1 Evaluation of the baseline project

4.1.1.1 Selection of subjects

The evaluation was divided into interviews and questionnaires. In order to complete the research subjects to the interviews were chosen. The interviews were held with four developers, and three project managers. The developers and one project manager were using Scrum as a development methodology and the other two project managers had never used Scrum. The reason for interviewing both developers and project managers was to get a view from both developers and project managers. The reason for interviewing everyone involved in the project was to get an idea of what everyone knew about Scrum and how the project was carried out regarding the development process. At first the view was that the entire project was run according to Scrum. The truth, that the project was only using parts of Scrum, became clearer after the interviews.

The questionnaire was given only to the developers in the baseline project and the new project.

4.1.1.2 Interviews

Interviews were held with both developers and project managers. The interviews consisted of gathering information of education, knowledge of different kinds of development processes (divided into project methodology and development methodology in the baseline project) what works in the project. These interview were made in order to get a view of what project methodology and development methodology was used, what the developers thought of the way the project was carried out, and how the development team and the project manager worked in parts of Scrum.

4.1.1.3 Questionnaire

After the interviews the developers received a questionnaire. This questionnaire consisted of questions of their knowledge of different phases in Scrum. The developers were not told that the questions were related to Scrum. The first 18 questions had a scale from 1-4, the next six questions had a scale from 1-5 and the last two questions had a scale from 1-5. Each question had to be answered in two areas. The first area was to get view of how much the baseline project used the particular practise. The second area was to get a view of how the developer would feel about using this practise to the full extent. The questionnaire consisted also of questions of their knowledge in different development phases (analysis, design, implementation, test and documentation).

This questionnaire was made to find out if the developers development process was similar to Scrum and how they would feel of using Scrum.
The second reason to the questionnaire was to measure the knowledge and view of Scrum at the developers.

4.2 Measurements

In order to compare the two projects, metrics were used to determine the size and complexity of the project. Metrics both from the project and the source code are presented.

The metrics from the projects are:

- *Number of team members*. Number of people in the project.
- *Measured time (Developers)*. Man hours spent in the project for the developers.
- *Measured time (Everybody)*. Man hours spent in the project for all involved.
- *Average estimate per Item*. The average time estimated on activities in the sprint backlog.
- *Min estimate*. The minimum estimated time on an activity in the sprint backlog.
- *Max Estimate*. The largest time estimated on an activity in the sprint backlog.
- *Median estimate*. The median of the time estimated in the sprint backlog.
- *Variance on estimate*. The variance between the time estimates on all activities in the sprint backlog.

To measure the code SourceMonitor Version 2.0 from Campwood Software [10] was used.

The metrics that were collected are:

- *Lines of Code (LOC)*. The total number of lines including comments.
- *Statements*. The number of statements in the program [14].
- *Classes*. Number of classes produced in the project.
- *Method per Class*. Total number of methods divided with total number of classes.
- *Calls per Method*. Total number of calls in the source code divided with the total number of methods.
- *Statements per Method*. Total number of statements divided with total number of methods.
- *Maximum block depth*. The deepest nested block depth level found in a class. Example on block depth three is

```java
class{
    function{
        statement {
        }
    }
}
```

- *Average block depth*. Average of nested block dept level in a class.
• **Percent comment line.** Percent of lines that begin with “//” or between “/\*….\*/” of total LOC.
• **Percent documentation line.** Percent of lines that begin with “///” of total LOC.

Only code that was included in the finished product and that was separate from already existing classes was measured. Code for testing is not included in the measurements.

To get a better overview of the metrics Kiviat diagrams are used to visualize the result.
5 ANALYSIS OF CASE STUDY

5.1 Metrics

These are the metrics from the baseline project and the new project. The estimate time in the new project is only for the developers and in the baseline project it also includes project manager and other resources.

<table>
<thead>
<tr>
<th></th>
<th>Baseline project</th>
<th>New project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated time</td>
<td>1122</td>
<td>652</td>
</tr>
<tr>
<td>Measured time (developers)</td>
<td>813</td>
<td>497</td>
</tr>
<tr>
<td>Measured time (Everybody)</td>
<td>993</td>
<td>508</td>
</tr>
<tr>
<td>Average number of Developers</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5.1 Time measurement from the two projects

As clearly illustrated in Figure 5.1 the new project is smaller than the baseline project. The implication on the study of this is that the communication and interaction between the team members and external resources will be different [8].

<table>
<thead>
<tr>
<th></th>
<th>Baseline project</th>
<th>New project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint Backlog Items</td>
<td>70</td>
<td>127</td>
</tr>
<tr>
<td>Average estimate per Item</td>
<td>10</td>
<td>5,3</td>
</tr>
<tr>
<td>Min estimate</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Max estimate</td>
<td>70</td>
<td>16</td>
</tr>
<tr>
<td>Median estimate</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Variance on estimate</td>
<td>140</td>
<td>12,5</td>
</tr>
</tbody>
</table>

Table 5.2 Measurements from all sprints
The variance on the estimated time is much smaller in the new project. The reason for this can be that the activities are divided into smaller scoops and thus are easier to estimate. The maximum estimated time in the new project is only 0.23 of the max estimate in the baseline project and the average time estimate is about half as large. A second reason can be that the estimation is done by the developers in a greater extent than in the baseline project where much more of the estimation is done by the project manager.

Table 5.3 Measurements from the source code

<table>
<thead>
<tr>
<th>Metric</th>
<th>Baseline project</th>
<th>New project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files</td>
<td>65</td>
<td>39</td>
</tr>
<tr>
<td>LOC</td>
<td>9869</td>
<td>4154</td>
</tr>
<tr>
<td>Statements</td>
<td>4943</td>
<td>1539</td>
</tr>
<tr>
<td>%Comments</td>
<td>15.2</td>
<td>9.1</td>
</tr>
<tr>
<td>%Docs</td>
<td>3</td>
<td>15.3</td>
</tr>
<tr>
<td>Classes</td>
<td>47</td>
<td>37</td>
</tr>
<tr>
<td>Method/Class</td>
<td>9.28</td>
<td>5.97</td>
</tr>
<tr>
<td>Calls/Method</td>
<td>6.72</td>
<td>2.33</td>
</tr>
<tr>
<td>Statements/Method</td>
<td>8.72</td>
<td>3.86</td>
</tr>
<tr>
<td>Max Complexity</td>
<td>39</td>
<td>13</td>
</tr>
<tr>
<td>Max Depth</td>
<td>9+</td>
<td>6</td>
</tr>
<tr>
<td>Avg Depth</td>
<td>2.87</td>
<td>1.55</td>
</tr>
</tbody>
</table>

Metrics related to source code were gathered from the two projects and is presented in table 5.3 and figure 5.3-4. Only source code that is included in the final product is measured. It should also be pointed out that the new project has more auto generated code than the baseline project due to a higher density of ASP.NET pages. In Appendix B the entire report generated by SourceMonitor is published. In the new project more definition and design work was done in the measured sprint than the baseline project. One can not conclude that the developers in the new project were
more productive than the developers in the baseline project, because of the difference in the two projects.

![Figure 5.3 Charts over the metrics collected by SourceMonitor](chart1)

**Figure 5.3** Charts over the metrics collected by SourceMonitor

![Figure 5.4 Kiviat graph for the baseline project and the new project](chart2)

**Figure 5.4** Kiviat graph for the baseline project and the new project

The Kiviat graphs in Figure 5.4 shows clear that both project have high complexity.

### 5.2 Case study

#### 5.2.1 Results from questionnaire

Since the new project did not have a project manager these answers refer to a project without a project manager. The result from the baseline project is the average from the four developers and in the new project from the two developers.

The scale represents these answers.

1 = I do not think it would be good.
Answers from Baseline project (1-4)  
Answers from new project (1-4)

Figure 5.5 Results from questionnaire

The size of the project has impact on the answers to question 5 and 10. If the team is not bigger than two persons as in the new project the communication between the group members is high and daily scrum meetings do not improve the awareness of what happens in the group. And a small project that do not need more than two or three people is not more effective if it is five to nine people involved.

Most of the developers were positive to use the key parts of Scrum more then they used them during the projects.

Figure 5.6 Results from questionnaire
15. How well do you document your work in the project (1 = too few, 2 = sufficient, 3 = too many)  
1.75  
2  
Table 5.4 Results from questionnaire

<table>
<thead>
<tr>
<th>Question</th>
<th>Answers from Baseline project (1-4)</th>
<th>Answers from new project (1-4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>How well do you know the design in your project?</td>
<td>3</td>
<td>3.5</td>
</tr>
<tr>
<td>How well do you know what other team members do in your project?</td>
<td>3.3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5.5 Results from questionnaire

Figure 5.6 Overall knowledge in the projects

The overall knowledge in the teams on the different phases of the development is similar. It should be in mind that the values are the developers own opinion of how good they are in the different arias and not a results from a test.

5.2.2 Results from interviews

This information is collected from the interviews with the developers in the baseline project and project manager at T-Systems Sweden. The management decided that the developers were going to use Scrum as a development method and Extreme Project Management was to be used as a project methodology. This reason for using
parts of Scrum as a development methodology was because the management decided that an agile approach to project is more effective than a non Agile. The project manager in the baseline project decided that Extreme Project Management should be used as a project methodology.

All developers in the baseline project felt that there was a lack of knowledge of the project methodology. An example on this was that they had different views of what they thought Scrum was. One of the developers thought that Scrum was a way of meeting every morning and another thought that Scrum was to deliver an increment each month.

There had not existed any information meetings at all for the developers about either the project- or the development methodology. The knowledge of the project and development methodology they had received was learned throughout the project. The developers that did know anything about the project methodology felt that the management could not decide on what to use and making new decisions about it all the time. One developer believed that it would be better to test a complete method instead of developing one at T-systems. Another result was that there has to be a way to find out what method that is being used and the method has to be taught to everyone on the project includes developers, business management and customers.

One way of solving the problem with informing the employees about the project methodology would be to have project methodology introduction meetings every year. In addition to that the project methodology should be documented and updated.

It is important for the customer to follow the project methodology. If the customer does not care, the developers automatically will not care to follow it.

Since it was the project manager and the technical advisor that set up the activities it was difficult for the developers to know what they really meant with the activities. The team solved it by talking to either the project manager or the technical advisor almost each time they were appointed an assignment. The manager and advisor should be more prepared on what assignments to distribute to the developers.

Every developer felt that the project manager worked for the group and not the other way around. The project manager helped with the larger problems i.e. talking to the customer, and the developers dealt with the smaller problems by themselves.

The developers did not know how many or which documents that were produced in the project. The general idea was that too few documents were produced. Developers need a definition document i.e. when someone else has to understand the code or for application management.

The scope on how the development should be carried out should be clearer.

The group size was excellent except that more resources were needed to cope with the work that should be completed.

The communication between the developers took place through MSN, or taking to directly to each other. A meeting in the middle of the Sprint was held for two hours which had good response from the team and the project manager. This meeting was held to gather information and inform the team and the project manager of which activities were completed.

If a developer has an internal meeting with either the project manager or the technical advisor that affected the rest of the group it did not always been communicated to them what have been said at that particular meeting.

Due to that the developers did not know how iterative development worked they felt that the definition phase was going on throughout the development phase as well. They felt like it would be good to have clear lines between definition and development. The developers all agreed that iterative development was better than traditional waterfall.

One of the developers meant, “In the IT industry the customer often receives a product that he or she considers is not as initially intended. This is why it is important to meet with the customer during the development in order to get the customer’s opinion during the development and not after.” Another developer meant, “One thing
that is important in order to get an increment is to look for and find a good increment in the planning phase.”

One developer felt that it was important to have daily Scrums to plan his/her day, “The Daily Scrum meeting starts the day for me”.

The problems on these meetings could be that they intend to consist of things that should not be brought up there (i.e. solutions). The same developer said that he/she could not pick activities from the Sprint backlog as it was not daily updated.

These are the problems with Scrum identified in the interviews:

- The developers did not know which practices Scrum consisted of
- Finding a deliverable increment in each iteration can be a challenge
- Since the customer is involved in project during the development phase the developers feel that they do not know when the definition phase stops.
- It is difficult to prioritise the product and Sprint backlog since the customer feels that all requirements have the same prioritization.
- The members in the Scrum team are not expert in all areas which leads to that certain member automatically are put on certain areas. This make the team non self organized.
- T-systems works with experts. The interviews showed that there was issue on how to temporary include these experts in the Scrum team.

The interviews gave the research the view that a very small part of Scrum was used in the baseline project. The members of the team knew very little of what project and development methodology that was applied in the project. The reason for that is that no education of which methodology that was applied had been given. No documents of what was applied had been written. The problem with the clear line between definition and development was based on the lack of knowledge of what an agile approach really meant. Since the customer is agile development welcomes changes in the requirements throughout the project no clear line between definition and development can exist in this approach.

The difficulties in prioritizing the Sprint and Product backlog are based on that the customer do not fully understands the commitment to the Agile approach. The business point of view in the project is run with non agile approach that is to say firm requirements without prioritization. The difficulties with prioritizing the Sprint backlog are on the other hand based on the lack of knowledge of Scrum in the development team. Since the project manager is prioritizes the Sprint backlog it is difficult to decide what has to be completed first from a technical point of view.
6 FEASIBILITY STUDY

To test whether the assumptions in the case study was true a feasibility study was made. The goal of the study consisted of testing if Scrum can be fitted to T-systems Sweden and to test assumptions from the baseline project regarding Scrum in a new project. The results from this study are presented as following.

Issues regarding Scrum that were known prior to the first Sprint

- The developers did not know which practises Scrum consisted of
- Finding a deliverable increment in each iteration can be a challenge
- Since the customer is involved in project during the development phase the developers’ feel that they do not know when the definition phase stops.
- It is difficult to prioritise the product and Sprint backlog since the customer feels that all requirements have the same prioritization
- The members in the Scrum team are not expert in all areas which leads to that certain member automatically are put on certain areas. This make the team non self organized.
- T-systems works with experts. The interviews showed that there was issue on how to temporary include these experts in the Scrum team.

What to be tested in the first Sprint:

- If the developers know what methodology is applied, is there less confusion?
- If Scrum is better taught to the developers, does a issue regarding deliverable increments exists?
- Is there a confusion regarding the line between definition and development?
- Is there an issue regarding prioritization of the product and Sprint backlog?
- Can self organization be applied?
- How will the experts be planned, included and used?

As can be observed in table 6.1-3 all the sprints were shorter and fewer developers than Scrum recommend. The development tools were to a great extent Microsoft products.

6.1 Sprint 1

All issues from the baseline project were considered and gave rise to a document where the project methodology, roles, document and definition were defined. This document was named the project document and could be referred to and looked upon when questions were raised during the Sprint.

A few days into the Sprint the Daily Scrum meetings were removed. These were removed since the project only had two developers and the communication within the project was complete [20].

During the first part of the first Sprint one of the developers was appointed as Scrum Master but after a while the developers realized that a Scrum master was not needed since both developers were involved in all parts of the project.

The first day of the Sprint the customer wanted a document where the requirements were explained in detail and a Use case diagram showed all actors in the
web application. Since this document was not planned the customer had to remove activities from the Sprint backlog in order to make room for this document.

The data in table 6.1 shows that Sprint 1 had an abnormal termination and lasted 10 days instead of 14. The cause for the abnormal termination was that the development environment was not setup until day nine. The reason for the late setup of the development environment was that the external resource responsible for setting it up was overbooked; even if all development tools were installed in the beginning of the sprint could no development begin.

The first sprint according to The Standish group definition where failed due to the termination of the sprint.

<table>
<thead>
<tr>
<th>Setup for Sprint 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Duration</strong></td>
</tr>
<tr>
<td><strong>Developers</strong></td>
</tr>
<tr>
<td><strong>External developer</strong></td>
</tr>
<tr>
<td><strong>Tools in Sprint</strong></td>
</tr>
<tr>
<td><strong>Development tools</strong></td>
</tr>
<tr>
<td><strong>Programming language</strong></td>
</tr>
</tbody>
</table>

Table 6.1 Data from Sprint 1

### 6.2 Sprint 2

The retrospective meeting from the first Sprint made the second Sprint more effective. The decisions from the retrospective meeting were:

- All documents that are produced should be written by one person and reviewed by the other
- The project document had a purpose and should still be used in the second Sprint
- The late setup of the development environment was the reason for the abnormal termination of Sprint 1
- The solution of the low prioritising of the project was that the developers should plan meeting more in advance.
- The day of the Sprint (1, 2, 3, etc) in the Sprint was replaced with the date. This change was made to easier understand what day to report on.
- Activities were sometimes too small (less than 4 hours). These activities were often more difficult to report and do. This was thus a something to remember at the Sprint planning meeting.

When the design was made in the second Sprint the implementation started. During the Second Sprint these issues were found:

- Originator was not needed and used in the Sprint backlog and thus removed.
- All refactoring should be done in the end of each Sprint and not after each activity.
- Although the invitation for the review meeting were sent long time prior to the meeting, the involved could not participate. This was due to that the new project was low prioritized. The review meeting was
held for one hour and all invited and needed in this meeting could not come.

The second sprint according to The Standish group definition was Successful, all the features and functions specified were finished in time.

<table>
<thead>
<tr>
<th>Setup for Sprint2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Duration</strong></td>
</tr>
<tr>
<td><strong>Developers</strong></td>
</tr>
<tr>
<td><strong>External</strong></td>
</tr>
<tr>
<td><strong>developer</strong></td>
</tr>
<tr>
<td><strong>Tools in Sprint</strong></td>
</tr>
<tr>
<td><strong>Development</strong></td>
</tr>
<tr>
<td><strong>Programming</strong></td>
</tr>
</tbody>
</table>

Table 6.2 Data from Sprint 2

6.3 **Sprint 3**

The retrospective meeting after Sprint 2 produced the following:
- Since a long and well performed planning meeting was held, the second Sprint produced what it should. Although there were impediments none of those led to delay in the Sprint.
- The developers felt that since there was more knowledge and experience of Scrum in the second Sprint, the work was more effective and smooth.
- Since the planning meeting in the third Sprint was shortened, the time estimations in the third Sprint were too big.

In the third sprint a web based tool called ScrumWorks[23] where used for manage the sprint backlog. This proved more effective than the use of MS excel, mostly due to the easiness for the developer to report the time left on the activities. The drawbacks were that it did not provide all the information and were not as flexible as an excel sheet.

During the third sprint no issues regarding the project methodology could be noticed.

The third sprint according to The Standish group definition was successful, all the features and functions specified were completed in time.

<table>
<thead>
<tr>
<th>Setup for Sprint3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Duration</strong></td>
</tr>
<tr>
<td><strong>Developers</strong></td>
</tr>
<tr>
<td><strong>External</strong></td>
</tr>
<tr>
<td><strong>developer</strong></td>
</tr>
<tr>
<td><strong>Tools in Sprint</strong></td>
</tr>
<tr>
<td><strong>Development</strong></td>
</tr>
<tr>
<td><strong>Programming</strong></td>
</tr>
</tbody>
</table>

Table 6.3 Data from Sprint 3
6.4  **Strengths in Scrum**

The largest strength in Scrum is that it focuses on efficiency. As it was identified in the literature study, the unnecessary documents are not created during projects following the Scrum methodology. Since in Scrum all phases of developing the product occur in the team, no teams are forced to wait for other teams to complete specific phases in the development. This is both proven in the literature study and was learned in the feasibility study. Unnecessary waiting for different phases to be completed is left out. The customer involvement is set to a level so that he/she can change direction of the project when necessary. According to Scrum, the customer has to be actively responsible of the requirements and this leads to that only the most important requirements for the customer are developed. In the feasibility study this lead to that all requirements were not developed and this kept the production cost at a low level.

6.5  **Weaknesses in Scrum**

To be able to use Scrum to the full extent in a company, that particular company has to be structured in a specific way. One of the important things in Scrum, according to the literature study is that the projects have to be so large that at least five people can be involved in the project throughout the development time. To set up these kinds of teams can be difficult in small companies. As seen in the new project it can work with two developers as well. The weakness is then that it can be difficult to find two developers that behold the knowledge to by themselves develop the product. This problem was at T-systems Sweden solved by using experts to assist the teams.

Since Scrum does not struggle to report any worked time per activity and this can be seen as a problem regarding follow-up statistics. In the new project the developers decided to report that anyway for the purpose of thesis research even though it was against the principles of Scrum. This was decided in order to get statistics for the metrics.
7 CONCLUSIONS

This paper addresses Scrum and how to fit Scrum to T-systems Sweden. To construct a solution to T-systems a literature study, case study and feasibility study has been made. Issues regarding Scrum in a baseline project have been identified and presented.

Based on the feasibility study we found that Scrum has to be modified to fit T-systems Sweden. Since the new project only consisted of two developers, the Scrum master and Scrum daily meetings were unnecessary and thus removed. The Scrum master’s role was unnecessary since all information and practices in the project were made by the Scrum team together or in close contact with each other. The resource lost was insignificant when both developers attended all the meetings. Since both developers knew everything in the project the daily meetings were unnecessary. If a Scrum master would be appointed to the project this would lead to those daily meetings had to be used to inform the Scrum master. One impediment when using Scrum is that developers in T-systems do not have equal high knowledge in all areas. A solution to this is to use expert in certain areas in certain moments of the project. We found out that planning these in the beginning and the end of the project was a solution that fit T-systems; not to include them in the project but to use them as an external resource fit this solution as well. A weakness in this solution is that extra information meetings are needed to inform these experts.

Tools that support the development process facilitate the information flow for the development team and management [22][21]. An Excel file is not suitable for an effective management of the backlog and task in the Scrum process when developers exceed two developers. This because excel do not allow more than one user to have writing rights to the document. A Scrum process application facilitates this and was therefore used in the third Sprint.

In the new project the first Sprint had to abnormally terminate and the reason for this was that the development environment was not set up until the middle of the Sprint. A conclusion of this was that if a Sprint consists of development this environment should be set up before the start of the Sprint.

The different role’s opinion of Scrum differed. The project managers felt that a formal method is important and the project manager’s experience is important when choosing that certain method. The developers felt that documentation and information of the method is very important in order for them to know what method that is being applied.

At T-systems only the project managers had got necessary education about Scrum and Extreme project Management. In order to use Scrum fully everyone in the project has to know how Scrum works. Education and documentation were two important issues regarding the development processes that were brought up in the quantitative research.

The benefits to use Scrum on T-systems are that a formal methodology increases the chance of success in projects. Since T-systems do not use a formal methodology a implementation of Scrum would increase the chance of success in the projects with 6 percent [4]. Since Scrum and Agile processes also consists of more executive support (support from customer) and minimized scope (iterative development) chances for success can rise with up to 34 percent [4]. This makes Scrum a strength factor in developing software.
8  FUTURE STUDIES

8.1  Bigger Teams
To study a Scrum team that is between 5-9 team members is interesting, both a team that are starting to use Scrum and a team that have used it for some time. How does the different team members in the team act and how they compliment each other. Do the team members always have a task to work on even in the beginning and end of a sprint?
How does scrum of scrums work and what do you win and what do you lose? Is it really good to split up a team of ten to two teams of five members?

8.2  Customer and stakeholders
What is the customer’s and stakeholders view of Scrum and do they believe that it is better than traditional software development. Do they get what they want more often than before?

8.3  Different kinds of products
How does Scrum work in development in different kinds of products? Is it as good to use in game development as in embedded system or in web application development etc. What is the difference in using scrum to develop a product to a mass market or to a customer?
## GLOSSARY

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activities</td>
<td>The functionality that is meant to be performed either in the Sprint (Sprint Backlog) or in the whole project (Project Backlog)</td>
</tr>
<tr>
<td>Chicken</td>
<td>All the stakeholders in the project. Everybody that are active in the project but aren’t developing. Everyone except the Scrum team, Scrum master and the Product owner.</td>
</tr>
<tr>
<td>Face-to-face communication</td>
<td>Communication should be performed through face-to-face and not through documentation.</td>
</tr>
<tr>
<td>Increment</td>
<td>Everything that is produced during a Sprint</td>
</tr>
<tr>
<td>Iteration</td>
<td>A time cycle. In Scrum it is called a Sprint</td>
</tr>
<tr>
<td>Pig</td>
<td>Someone who actually performs the work in a project, that is the Product Owner, Scrum team and the Scrum master.</td>
</tr>
<tr>
<td>Product Backlog</td>
<td>A constantly updated and prioritised activity list that contain function that be added to the project</td>
</tr>
<tr>
<td>Product Burndown Chart</td>
<td>A graph showing how much is to be done in the project</td>
</tr>
<tr>
<td>Product Owner</td>
<td>The person that is responsible for the product backlog. Is seen as the person that is responsible for the project as a whole. This person is the usually the customer. In “in House” project this person can also be the Scrum master.</td>
</tr>
<tr>
<td>Review meetings</td>
<td>In the end of each Sprint the stakeholders come together in a meeting to see what the Scrum team has produced in the latest Sprint.</td>
</tr>
<tr>
<td>Scrum master</td>
<td>replaces the ordinary project manager and is responsible to make sure to remove any obstacles in the project. Also works as a firewall between the management and the Scrum team.</td>
</tr>
<tr>
<td>Daily Scrum meetings</td>
<td>Morning meetings that should last for 15 minutes. The Scrum team members inform each other of the past day, the current day and any obstacles that could prevent them from doing their work.</td>
</tr>
<tr>
<td>Scrum team</td>
<td>The development team that develops the product</td>
</tr>
<tr>
<td>Sprint</td>
<td>A time cycle that last for 30 calendar days</td>
</tr>
<tr>
<td>Sprint Backlog</td>
<td>A constantly updated list of activities that should be completed during the Sprint.</td>
</tr>
<tr>
<td>Sprint Burndown Chart</td>
<td>A graph that shows how much the activities take that are left in the Sprint</td>
</tr>
<tr>
<td>Sprint Goal</td>
<td>During the Sprint Planning meeting the Goal with the Sprint is set. This is called the Sprint Goal.</td>
</tr>
<tr>
<td>Sprint Planning meeting</td>
<td>On the first day of the Sprint two planning meetings are set up. During these meeting the next Sprint is planned.</td>
</tr>
<tr>
<td>Sprint retrospective meeting</td>
<td>After the review meeting a meeting is set up for the Scrum team. During this meeting the Scrum team discusses how to make the next Sprint more effective.</td>
</tr>
</tbody>
</table>
REFERENCES

11  **APPENDIX A. QUESTIONNAIRE**

Enkätfrågor
Denna enkät innehåller 25 frågor och svaren kommer givetvis inte att visas för andra. Frågorna är ställda för att ta reda på vad som används i projektet, vad du tycker samt hur bra du anser dig vara på olika områden inom utveckling.

Vad används och vad tycker du?

Dessa frågor ställs för att vi ska veta uppfattningen om hur saker och ting fungerar inom projektgruppen.

1. **Iterativ utveckling**
   *Innehär att en iterationen pågår i 30 dagar*

   Hur anser du detta tillämpas i Boomerang CRM fas 2?

<table>
<thead>
<tr>
<th>Vi använder oss inte av detta</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Vi ska använda det men gör det inte</td>
<td></td>
</tr>
<tr>
<td>Vi använder det delvis</td>
<td></td>
</tr>
<tr>
<td>Vi använder det</td>
<td></td>
</tr>
</tbody>
</table>

   Vad är din uppfattning om detta skulle tillämpas fullt ut?

   | Jag tycker inte att det alls är bra |   |
   | Jag tycker att det är mindre bra |   |
   | Jag tycker att det är rätt bra |   |
   | Jag tycker att det är utmärkt |   |

2. **Sprint review**
   *Innehär att där arbetsgruppen presenterar en del av produkten för kunden och ledningen på företaget*

   Hur anser du detta tillämpas i Boomerang CRM fas 2?

   | Vi använder oss inte av detta |   |
   | Vi ska använda det men gör det inte |   |
   | Vi använder det delvis |   |
   | Vi använder det |   |

   Vad är din uppfattning om detta skulle tillämpas fullt ut?

   | Jag tycker inte att det alls är bra |   |
   | Jag tycker att det är mindre bra |   |
   | Jag tycker att det är rätt bra |   |
   | Jag tycker att det är utmärkt |   |

3. **Prioriteringen i produkt backlogen**
Det är kunden som prioriterar i produkt backlogen och den omprioriteras under projektets gång

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

- Vi använder oss inte av detta  
- Vi ska använda det men gör det inte  
- Vi använder det delvis  
- Vi använder det  

Vad är din uppfattning om detta skulle tillämpas fullt ut?

- Jag tycker inte att det alls är bra  
- Jag tycker att det är mindre bra  
- Jag tycker att det är rätt bra  
- Jag tycker att det är utmärkt  

4. Prioriteringen i Sprint backlogen

Det är teamet och projektledaren som bestämmer prioriteringen i Sprintbacklogen.

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

- Vi använder oss inte av detta  
- Vi ska använda det men gör det inte  
- Vi använder det delvis  
- Vi använder det  

Vad är din uppfattning om detta skulle tillämpas fullt ut?

- Jag tycker inte att det alls är bra  
- Jag tycker att det är mindre bra  
- Jag tycker att det är rätt bra  
- Jag tycker att det är utmärkt  

5. SCRUM morgonmöten

Möte varje morgon där alla utvecklare träffas (15-30 minuter).

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

- Vi använder oss inte av detta  
- Vi ska använda det men gör det inte  
- Vi använder det delvis  
- Vi använder det  

Vad är din uppfattning om detta skulle tillämpas fullt ut?

- Jag tycker inte att det alls är bra  
- Jag tycker att det är mindre bra  
- Jag tycker att det är rätt bra  
- Jag tycker att det är utmärkt  

42
6. **Face-to-face kommunikation**  
*All kommunikation sker face-to-face och inte via dokumentation.*

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

- Vi använder oss inte av detta
- Vi ska använda det men gör det inte
- Vi använder det delvis
- Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?

- Jag tycker inte att det alls är bra
- Jag tycker att det är mindre bra
- Jag tycker att det är rätt bra
- Jag tycker att det är utmärkt

7. **Att teamet sitter i öppna kontorslandskap bredvid varandra**

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

- Vi använder oss inte av detta
- Vi ska använda det men gör det inte
- Vi använder det delvis
- Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?

- Jag tycker inte att det alls är bra
- Jag tycker att det är mindre bra
- Jag tycker att det är rätt bra
- Jag tycker att det är utmärkt

8. **Gruppen sammansätts så att gruppmedlemmarna har bred kunskap inom många kunskapsområden.**

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

- Vi använder oss inte av detta
- Vi ska använda det men gör det inte
- Vi använder det delvis
- Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?

- Jag tycker inte att det alls är bra
- Jag tycker att det är mindre bra
- Jag tycker att det är rätt bra
- Jag tycker att det är utmärkt

9. **Lära varandra om olika kunskapsområden.**
Få hjälp av andra i teamet under Sprinten om hur man kan lösa problem

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

Vi använder oss inte av detta
Vi ska använda det men gör det inte
Vi använder det delvis
Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?

Jag tycker inte att det alls är bra
Jag tycker att det är mindre bra
Jag tycker att det är rätt bra
Jag tycker att det är utmärkt

10. 5-9 utvecklare per projekt

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

Vi använder oss inte av detta
Vi ska använda det men gör det inte
Vi använder det delvis
Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?

Jag tycker inte att det alls är bra
Jag tycker att det är mindre bra
Jag tycker att det är rätt bra
Jag tycker att det är utmärkt

11. Löpande kommunikation med kunden

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

Vi använder oss inte av detta
Vi ska använda det men gör det inte
Vi använder det delvis
Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?

Jag tycker inte att det alls är bra
Jag tycker att det är mindre bra
Jag tycker att det är rätt bra
Jag tycker att det är utmärkt

12. Återanvändning av kod internt
Få kod från andra gruppmmedlemmar

Hur anser du att detta tillämpas i Boomerang CRM fas 2?
Vi använder oss inte av detta
Vi ska använda det men gör det inte
Vi använder det delvis
Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?
Jag tycker inte att det alls är bra
Jag tycker att det är mindre bra
Jag tycker att det är rätt bra
Jag tycker att det är utmärkt

13.
Återanvändning av kod externt
Få kod från tidigare projekt

HUR ANSER DU ATT DETTA TILLÄMPAS I BOOMERANG CRM FAS 2?
Vi använder oss inte av detta
Vi ska använda det men gör det inte
Vi använder det delvis
Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?
Jag tycker inte att det alls är bra
Jag tycker att det är mindre bra
Jag tycker att det är rätt bra
Jag tycker att det är utmärkt

14.
Berätta för projektledaren om problem som du stöter på i projektet.

HUR ANSER DU ATT DETTA TILLÄMPAS I BOOMERANG CRM FAS 2?
Vi använder oss inte av detta
Vi ska använda det men gör det inte
Vi använder det delvis
Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?
Jag tycker inte att det alls är bra
Jag tycker att det är mindre bra
Jag tycker att det är rätt bra
Jag tycker att det är utmärkt

15.
Dokumentationen

Vad är din uppfattning om detta?
Jag tycker att vi har för lite dokumentation
Jag tycker att vi har lagom med dokumentation
Jag tycker att vi har för mycket dokumentation

### 16. Projektledaren arbetar för gruppen och inte tvärtom.

*Det är i själva verket arbetsgruppen som bestämmer under en Sprint och projektledaren arbetar för att tillgodose arbetsgruppens krav.*

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

- Vi använder oss inte av detta
- Vi ska använda det men gör det inte
- Vi använder det delvis
- Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?

- Jag tycker inte att det alls är bra
- Jag tycker att det är mindre bra
- Jag tycker att det är rätt bra
- Jag tycker att det är utmärkt

### 17. Gruppen är självständig och kan inte bli störd av företagsledningen eller kund

*När Sprinten är igång får ingen utanför teamet bestämma något om hur arbetet ska gå till samt att ingen får ändra Sprint Backlogen.*

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

- Vi använder oss inte av detta
- Vi ska använda det men gör det inte
- Vi använder det delvis
- Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?

- Jag tycker inte att det alls är bra
- Jag tycker att det är mindre bra
- Jag tycker att det är rätt bra
- Jag tycker att det är utmärkt

### 18. Gruppen är självorganiserad?

*Gruppen organiserar sig själv och bestämmer vem som ska göra vad.*

Hur anser du att detta tillämpas i Boomerang CRM fas 2?

- Vi använder oss inte av detta
- Vi ska använda det men gör det inte
- Vi använder det delvis
- Vi använder det

Vad är din uppfattning om detta skulle tillämpas fullt ut?

- Jag tycker inte att det alls är bra
- Jag tycker att det är mindre bra
Jag tycker att det är rätt bra
Jag tycker att det är utmärkt

Din kunskap
Här skulle vi vilja veta hur bra du anser dig vara inom olika områden i produktutvecklingen.

19. 
**Hur bra anser du att din kunskap är inom Analys?**

Kan inget
Kan lite
Kan relativt bra
Är rätt duktig
Kan allt

20. 
**Hur bra anser du att din kunskap är inom design?**

Kan inget
Kan lite
Kan relativt bra
Är rätt duktig
Kan allt

21. 
**Hur bra anser du att din kunskap är inom programmering?**

Kan inget
Kan lite
Kan relativt bra
Är rätt duktig
Kan allt

22. 
**Hur bra anser du att din kunskap är inom testning?**  
*Både skrivning av testfall samt testning.*

Kan inget
Kan lite
Kan relativt bra
Är rätt duktig
Kan allt

23. 
**Hur bra anser du att din kunskap är inom dokumentation?**

Kan inget
Kan lite
Kan relativt bra
Är rätt duktig
Kan allt
Övrigt

24. Hur insatt är du i designen i CRM projektet?
   Inte alls  □
   Lite  □
   Relativt insatt  □
   Väl insatt  □

25. Hur väl insatt är du i vad andra i teamet håller på med i CRM projektet?
   Inte alls  □
   Lite  □
   Relativt insatt  □
   Väl insatt  □

Vi vill tacka för ditt deltagande.

Lior och Marcus
# Appendix B. Measurement Data from SourceMonitor

Metrics Summary For Checkpoint 'Baseline' [Printed on 15 jun 2005]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Directory</td>
<td>C:\tmp\CRM</td>
</tr>
<tr>
<td>Project Name</td>
<td>CRM_2</td>
</tr>
<tr>
<td>Checkpoint Name</td>
<td>Baseline</td>
</tr>
<tr>
<td>Created On</td>
<td>15 juni 2005, 09:54:11</td>
</tr>
<tr>
<td>Files</td>
<td>40</td>
</tr>
<tr>
<td>Lines</td>
<td>6,151</td>
</tr>
<tr>
<td>Statements</td>
<td>2,969</td>
</tr>
<tr>
<td>Percent Comment Lines</td>
<td>13.6</td>
</tr>
<tr>
<td>Percent Documentation Lines</td>
<td>4.1</td>
</tr>
<tr>
<td>Classes, Interfaces, Structs</td>
<td>34</td>
</tr>
<tr>
<td>Methods per Class</td>
<td>10.03</td>
</tr>
<tr>
<td>Calls per Method</td>
<td>4.32</td>
</tr>
<tr>
<td>Statements per Method</td>
<td>6.38</td>
</tr>
<tr>
<td>Line Number of Most Complex Method</td>
<td>(undefined)</td>
</tr>
<tr>
<td>Name of Most Complex Method</td>
<td>Reader-using()</td>
</tr>
<tr>
<td>Maximum Complexity</td>
<td>32</td>
</tr>
<tr>
<td>Line Number of Deepest Block</td>
<td>(undefined)</td>
</tr>
<tr>
<td>Maximum Block Depth</td>
<td>9</td>
</tr>
<tr>
<td>Average Block Depth</td>
<td>2.53</td>
</tr>
</tbody>
</table>

Names of Methods in 29 Class(es): Complexity

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGC-handler.HandlePermissionErrorCodes()</td>
<td>20</td>
</tr>
<tr>
<td>Class1.Main()</td>
<td>1</td>
</tr>
<tr>
<td>ClosingRecord.FromString()</td>
<td>2</td>
</tr>
<tr>
<td>Common.Person.Number()</td>
<td>7</td>
</tr>
<tr>
<td>CreditRecord.FromString()</td>
<td>2</td>
</tr>
<tr>
<td>CustomerRecord.FromString()</td>
<td>2</td>
</tr>
<tr>
<td>DCH-file.GetSettings()</td>
<td>20</td>
</tr>
<tr>
<td>ErrorRecord.ErrorRecord()</td>
<td>1</td>
</tr>
<tr>
<td>Form1.Dispose()</td>
<td>3</td>
</tr>
<tr>
<td>Handler.ExecuteScalar()</td>
<td>3</td>
</tr>
<tr>
<td>InvoiceRecord.FromString()</td>
<td>2</td>
</tr>
<tr>
<td>OCRCustomerRecord.OCR.CustomerRecord()</td>
<td>1</td>
</tr>
<tr>
<td>OCROpeningRecord.OCR.OpeningRecord()</td>
<td>1</td>
</tr>
<tr>
<td>OCRProcessingDateRecord.OCR.ProcessingDateRecord()</td>
<td>1</td>
</tr>
<tr>
<td>OCRRecipientRecord.OCR.RecipientRecord()</td>
<td>1</td>
</tr>
<tr>
<td>OCRSubTotalRecord.OCR.SubTotalRecord()</td>
<td>1</td>
</tr>
<tr>
<td>OCRTotalRecord.OCR.TotalRecord()</td>
<td>1</td>
</tr>
<tr>
<td>OCRTransactionRecord.OCR.TransactionRecord()</td>
<td>1</td>
</tr>
<tr>
<td>OpeningRecord.FromString()</td>
<td>2</td>
</tr>
<tr>
<td>PaymentChangeRecord.FromString()</td>
<td>3</td>
</tr>
<tr>
<td>PaymentRecord.FromString()</td>
<td>3</td>
</tr>
<tr>
<td>PermissionRecord.FromString()</td>
<td>3</td>
</tr>
<tr>
<td>QueueItem.FixConnectionString()</td>
<td>4</td>
</tr>
<tr>
<td>Reader-using()</td>
<td>32</td>
</tr>
<tr>
<td>ReceiverData.ReceiveData()</td>
<td>1</td>
</tr>
<tr>
<td>Record.ReadFIN()</td>
<td>8</td>
</tr>
<tr>
<td>RecordCollection.getFiner()</td>
<td>1</td>
</tr>
<tr>
<td>SubscriptionInvoice.init()</td>
<td>9</td>
</tr>
<tr>
<td>Writer.WriteFile()</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Block Depth</th>
<th>Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>226</td>
</tr>
<tr>
<td>1</td>
<td>344</td>
</tr>
<tr>
<td>2</td>
<td>1,159</td>
</tr>
<tr>
<td>3</td>
<td>650</td>
</tr>
<tr>
<td>4</td>
<td>337</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>5</td>
<td>122</td>
</tr>
<tr>
<td>6</td>
<td>31</td>
</tr>
<tr>
<td>7</td>
<td>24</td>
</tr>
<tr>
<td>8</td>
<td>26</td>
</tr>
<tr>
<td>9+</td>
<td>40</td>
</tr>
</tbody>
</table>
### Metrics Summary For Checkpoint 'New Project' (Printed on 15 jun 2005)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Directory</td>
<td>C:\tmpLAP/</td>
</tr>
<tr>
<td>Project Name</td>
<td>New Project</td>
</tr>
<tr>
<td>Checkpoint Name</td>
<td>New Project</td>
</tr>
<tr>
<td>Created On</td>
<td>15 jun 2005, 19:47:16</td>
</tr>
<tr>
<td>Files</td>
<td>39</td>
</tr>
<tr>
<td>Lines</td>
<td>4,154</td>
</tr>
<tr>
<td>Statements</td>
<td>1,539</td>
</tr>
<tr>
<td>Percent Comment Lines</td>
<td>9.1</td>
</tr>
<tr>
<td>Percent Documentation Lines</td>
<td>15.3</td>
</tr>
<tr>
<td>Classes, Interfaces, Structs</td>
<td>37</td>
</tr>
<tr>
<td>Methods per Class</td>
<td>5.97</td>
</tr>
<tr>
<td>Cells per Method</td>
<td>2.33</td>
</tr>
<tr>
<td>Statements per Method</td>
<td>3.66</td>
</tr>
<tr>
<td>Line Number of Most Complex Method</td>
<td>(undefined)</td>
</tr>
<tr>
<td>Name of Most Complex Method</td>
<td>uscdetditemBase.getItems()</td>
</tr>
<tr>
<td>Maximum Complexity</td>
<td>13</td>
</tr>
<tr>
<td>Line Number of Deepest Block</td>
<td>(undefined)</td>
</tr>
<tr>
<td>Maximum Block Depth</td>
<td>6</td>
</tr>
<tr>
<td>Average Block Depth</td>
<td>1.55</td>
</tr>
</tbody>
</table>

---

### Names of Methods in 34 Class(es):

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>busBaseGet</td>
<td>1</td>
</tr>
<tr>
<td>busGetCustomer</td>
<td>1</td>
</tr>
<tr>
<td>busGetItemBase</td>
<td>1</td>
</tr>
<tr>
<td>busGetItemContaining</td>
<td>-1</td>
</tr>
<tr>
<td>busGetItemLocatedAt</td>
<td>1</td>
</tr>
<tr>
<td>busListAndFoundDropDown</td>
<td>7</td>
</tr>
<tr>
<td>busSaveContents</td>
<td>1</td>
</tr>
<tr>
<td>busSaveCustomer.saveNewCustomer()</td>
<td>5</td>
</tr>
<tr>
<td>busSaveDetail</td>
<td>1</td>
</tr>
<tr>
<td>busSaveLocatedAt</td>
<td>1</td>
</tr>
<tr>
<td>busSearchItem</td>
<td>2</td>
</tr>
<tr>
<td>DBClass.DBClass()</td>
<td>1</td>
</tr>
<tr>
<td>facGetItem</td>
<td>1</td>
</tr>
<tr>
<td>facGetItemBase</td>
<td>1</td>
</tr>
<tr>
<td>facGetItemContents</td>
<td>1</td>
</tr>
<tr>
<td>facItemLocatedAt</td>
<td>1</td>
</tr>
<tr>
<td>facItemSearch</td>
<td>1</td>
</tr>
<tr>
<td>facSaveItem</td>
<td>1</td>
</tr>
<tr>
<td>uscContents</td>
<td>5</td>
</tr>
<tr>
<td>uscCustomerInformation</td>
<td>5</td>
</tr>
<tr>
<td>uscdetditemBase.getItems()</td>
<td>13</td>
</tr>
<tr>
<td>uscListContents.Page_Load()</td>
<td>1</td>
</tr>
<tr>
<td>uscListCustomerInformation.Page_Load()</td>
<td>1</td>
</tr>
<tr>
<td>uscListFile.buildMenu()</td>
<td>7</td>
</tr>
<tr>
<td>uscListLocatedAt</td>
<td>1</td>
</tr>
<tr>
<td>uscListCollocatedDoSave()</td>
<td>6</td>
</tr>
<tr>
<td>uscMultiDetailItem.GetUsCtrlIndex()</td>
<td>2</td>
</tr>
<tr>
<td>uscPopupContents.Page_Load()</td>
<td>1</td>
</tr>
<tr>
<td>uscPopupCustomerInfo.Page_Load()</td>
<td>1</td>
</tr>
<tr>
<td>uscPopupLocatedAt.Page_Load()</td>
<td>1</td>
</tr>
<tr>
<td>uscSearchItem.DoSearchAndFilter()</td>
<td>3</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Block Depth</th>
<th>Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Metrics Summary For Checkpoint 'New Project' (Printed on 15 Jun 2005)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>340</td>
</tr>
<tr>
<td>1</td>
<td>329</td>
</tr>
<tr>
<td>2</td>
<td>677</td>
</tr>
<tr>
<td>3</td>
<td>118</td>
</tr>
<tr>
<td>4</td>
<td>39</td>
</tr>
<tr>
<td>5</td>
<td>28</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>9+</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX C. QUESTIONS FOR THE INTERVIEWS

Introduktion

Anledningen till varför vi ska göra dessa intervjuer är att få oss en uppfattning till hur projektmetodiken fungerar, likheter och skillnader mot SCRUM samt er uppfattning om projekt/utvecklingsmetodiken. Vi kommer också att utreda rollers uppfattning.

Intervjuerna vi kommer att ta med er kommer att ta mellan 30 min och 1 timme. Vi kommer att skicka ut några av frågorna i förväg.

Vad vill vi ha ut av CRM studien?
• Veta hur deras utvecklingsmetodik fungerar
• Vad skiljer den från SCRUM
• Vad är likheterna med SCRUM
• Veta vad olika roller har för uppfattning av utvecklingsmetodiken
• Undersöka vad top management anser och jämföra denna information med andra roller i projektet
  • Identifiera olika roller
  • Hur kan man förhindra att hinder uppstår i framtida projekt? Samt föreslå förbättringar.
• Veta hur de olika faserna fungerar i Agile/SCRUM/T-systems.

Utbildning
1. Vilken utbildning har du?
2. Hur länge har du jobbat inom Itbranschen?
3. Hur länge har du jobbat på T-systems (ink primix)?
4. Hur ofta jobbar du på T-systems och ofta är du konsultad?
5. Vilka är dina arbetsuppgifter på företaget?
6. Vilka roller har du haft under din karriär?
7. Hur många timmar jobbar du ungefär per vecka?

Projektmetodik/utvecklingsmetodik
8. Vad känner du till om
  • T-frame
  • T-Rex
  • SCRUM
  • den nya projektmetodiken
  • eXtreme Project management
  • Flexible project Management
9. Vad hade du för utbildning/information innan ni började med den nya projektmetodiken?
10. Vad var din första uppfattning om den nya projektmetodiken?
11. Hur bra anser du att din kunskap är om den nu?

**CRM**

12. Vad innefattar din roll i projektet?
13. Gör du något som inte innefattar din roll?
15. Borde utdelning av uppgifter ske på annat sätt än hur det fungerar idag?
16. Anser du att du jobbar för projektledaren eller han för dig?
17. Finns det utsatta faser och i så fall vilka (Analys, design, kodning, testning)
18. Finns det någon standard för de dokument som skapas? (design, krav, tekniska)
19. Tycker du att det behövs?
20. Har ni någon kodstandard?
21. Tycker du att detta är bra (ha eller inte ha kodstandard)?
22. Anser du att det var är för mycket eller för lite dokumentation?
23. Vad anser du om att visa en delprodukt mellan sprintarna? (tar det onödig tid eller är det bra?)
24. Vad fungerar bra på morgonmötena anser du?
25. Vad kan förbättras på dessa möten?
26. Vad tycker du om gruppstorleken?
27. Vad gör du då du inte har något att göra?
28. Hur fungerar kommunikationen i gruppen? (fungerar det bra med möten eller skulle du vilja ha det på ett annat sätt?)
29. Vad tycker du om hur ni sitter när ni jobbar?
30. Hur anser du att man skulle kunna förbättra projektstrukturen i framtiden?
31. Skulle du sett fram emot att delta i ett projekt med en annan projektmetodik?