

Mikko Takkunen

Scrum Implementation In a Virtual Team Environment

Helsinki Metropolia University of Applied Sciences

Master's Degree In Business Administration

Business Informatics

11 November 2014

Author Title	Mikko Takkunen Scrum Implementation In a Virtual Team Environment
Number of Pages Date	44 pages + 2 appendices 11 November 2014
Degree	Master of Business Administration
Degree Programme	Master's Degree
Specialisation option	Business Informatics
Instructors	Pia Hellman, DSc (Econ) James Collins, PhD
<p>Following globalization, cost optimizing and rapidly developing technical resources, virtual teams are becoming more relevant in the contemporary workplace. Virtual teams present their own challenges which can make it difficult to analyze root causes for performance issues. Competence issues with individual team members might not always be the problem, as running virtual teams require different approaches than conventional, co-located teams.</p> <p>Virtual team was experiencing different performance problems which were hard to pinpoint exactly. Questionnaire and a survey were used to create a SWOT and help identify the problems in more detail. After this, a conceptual framework of virtual teams and different software development methods was studied to deepen understanding about the nature of the problems. An agile software development method, Scrum was identified as a tool that could improve the issues the team was having problems with. Scrum was implemented following strictly its rules and the team used it daily for approximately 4 months. After this period, the same survey that was used to identify the problems in the beginning was sent again. The team felt that all performance issue areas were improved after Scrum was introduced. The team will continue using Scrum and it is now being adopted to other business units as well.</p> <p>The study suggests that there are similarities between typical virtual team challenges and the strengths of using Scrum. The results of this thesis can be used to show that Scrum provides tools to increase virtual team performance.</p>	
Keywords	virtual teams, co-located teams, nearshoring, teamwork, scrum

Contents

1	Introduction	1
1.1	Research setting	1
1.2	Business problem and objectives	2
1.3	Scope and limitations	4
2	Research progress and methods	5
2.1	Steps and structure of the thesis	5
2.2	Research design	7
3	Current state analysis	8
3.1	Survey	8
3.1.1	Time related issues	9
3.1.2	Quality related questions	10
3.1.3	Operative issues	11
3.2	SWOT	12
4	Conceptual framework	13
4.1	Group or a team?	13
4.2	Virtual teams	14
4.3	Challenges in virtual teams	15
4.4	Enabling trust	16
4.5	Agile and traditional software development methods	17
4.5.1	Traditional software development	17
4.5.2	Agile development	18
4.6	Solving virtual team challenges and SWOT weaknesses with Scrum	24
5	Case study application	26
5.1	Implementing Scrum	27
5.2	Scrum calendar and workflow	27
5.3	Backlog	29
5.4	Scrum board	30
5.5	Sprint reports	31
5.6	Sprint planning	33
5.7	Instant messaging	34

6	Results	35
6.1	Time related issues	36
6.2	Quality related issues	37
6.3	Operative issues	38
7	Conclusions	39
	References	41
	Appendices	
	Appendix 1. Questionnaire on performance issues	
	Appendix 2. June and September Surveys	

1 Introduction

This thesis is a labour of both necessity and passion on understanding and developing teamwork. This thesis is a necessity – because it directly relates to my everyday work in business oriented IT and discusses how a particular problem was noticed and addressed. Business unit was experiencing quality and time problems in their development cycle as well as lack of trust. A study was made to identify the problem and then corrective actions taken by implementing a new software development method. This thesis is also an outcome of passion - because up to this day the greatest feelings of accomplishment, success and satisfaction I have gotten from all areas of life have had one thing in common. Interacting with my fellow people towards a common goal in a respectful, fair and mutually supportive way.

The objective of this thesis is to recognize performance problems within a virtual team in a case company, implement a solution to improve it and evaluate how well the implementation worked. The following research questions were created;

- 1) What performance issues does the virtual team have?
- 2) What is the best method to solve them?
- 3) Did the implementation of the chosen method help improve performance?

After introduction, the research progress and methods are presented followed by current state analysis, conceptual framework, case study application and ending with results and conclusions.

1.1 Research setting

Target company of the thesis is Oikotie Recruitment, the leading commercial service for distributing job ads in Finland. Oikotie Recruitment consists of three different services;

- 1) Printed job ads of the Helsingin Sanomat newspaper
- 2) Oikotie Työpaikat and Avoin Työpaikka, online classified job ad sites
- 3) Taito recruitment system

All services are run inside the Oikotie Recruitment business unit with each having their own team responsible of their everyday operations. Some members work across different services and some are dedicated for just the one service. Oikotie Recruitment has its own steering group which consists of key personnel from all services. Oikotie Recruitment is a part of Oikotie, which is a part of Sanoma News, a subsidiary in turn of the Sanoma Business Group. Turnover for Oikotie in 2013 was 13 million Euros. (Inoa, 2014)

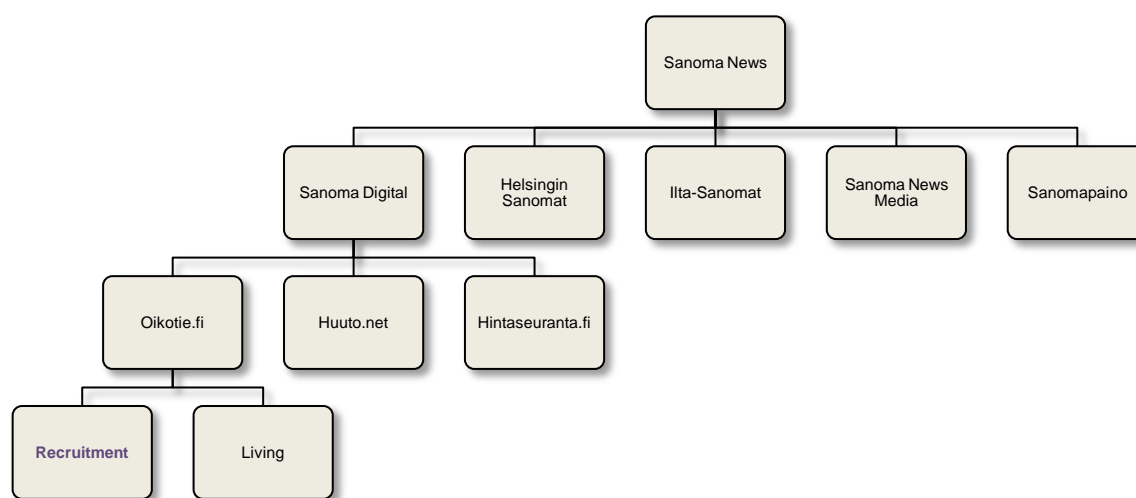


Figure 1. Corporate environment

For the thesis, a case is presented from one of the services under Oikotie Recruitment; the cloud based online recruitment system Taito. The team responsible for developing Taito consists of a Product Manager, Production Manager and a team of software developers. This service has one major difference to all other services; the software development team is nearshored to Poland, and all interaction is done via instant messaging, video- and teleconferencing.

1.2 Business problem and objectives

In a virtual team, where the product and technical managers were located in Finland and the rest of the development team located in Poland, the Finnish managers felt that there were performance issues related to the output of the Polish team. Pinpointing the problems was difficult as there were no mutually agreed, objective metrics and limited daily interaction. Atmosphere of mistrust was hanging between the managers in

Finland and the developers in Poland. Using the Collins’ 5 Golden Q & A’s of describing a research problem (Collins, 2012), we can summarise the following;

- The issue is that in order to be competitive, the development cycle from an idea to a working and deliverable function must be as fast and error free as possible.
- This is important because the impact of the issue in the short run is that the company will lose money and opportunity if it can not react fast enough to deliver relevant services for the customers and in the long run the current service will most likely end as obsolete
- Given this, the objective of the study will be to identify and solve problems within the development iteration loop to make it as efficient as possible
- The output of the study will be an implementation of a development method to improve the observed problems and evaluating how well the implementation worked
- This helps the organization because by identifying and solving these problems will enable the viability and monetization of the service

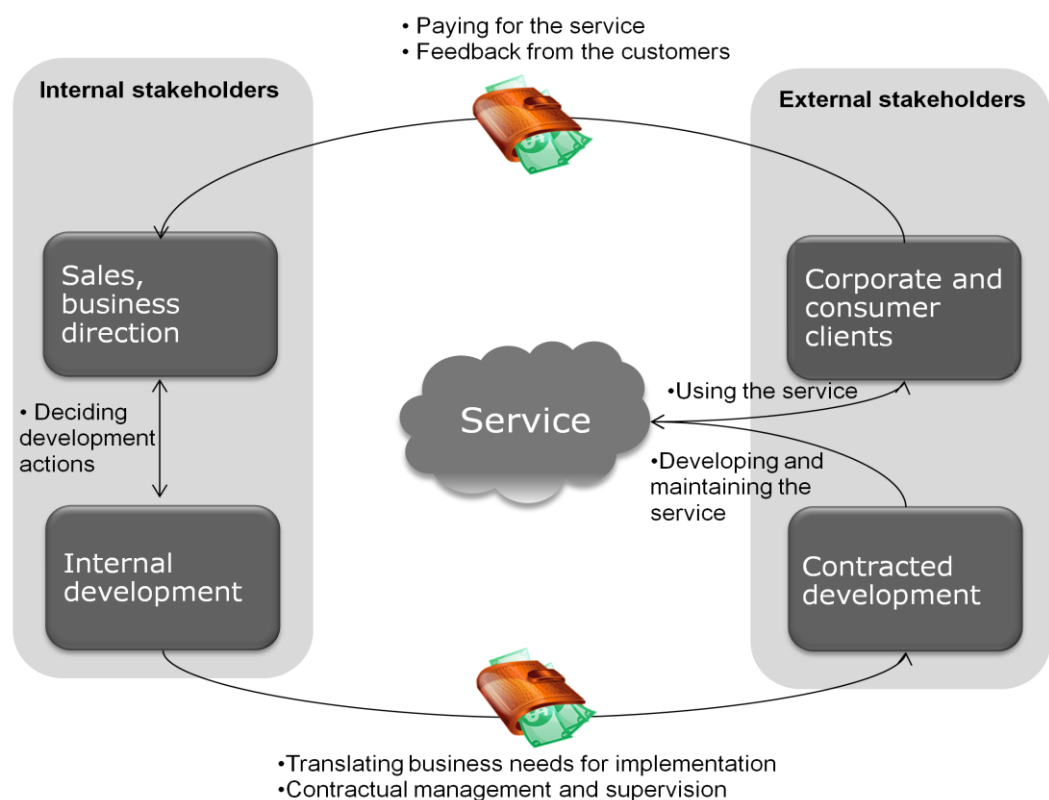


Figure 2. Thesis context

1.3 Scope and limitations

Even though the presented findings from the literature will be applicable in other contexts than the presented case study alone, the presented solution cannot be cloned directly for general use. It is noteworthy that the scope of the thesis does not handle all possible aspects of the service; for example sales, marketing and customer support are not considered in this thesis to play any part, even though in the actual business environment they play a major role on the overall success of the service. The nature of the study is such that it does not explore the phenomena from all possible angles. Industry wide, unified metrics for the measured issues such as time, quality and trust are not investigated as part of this thesis.

2 Research progress and design

2.1 Steps and structure of the thesis

The thesis is constructed from six different steps with each having their own outputs to be used in the following step.

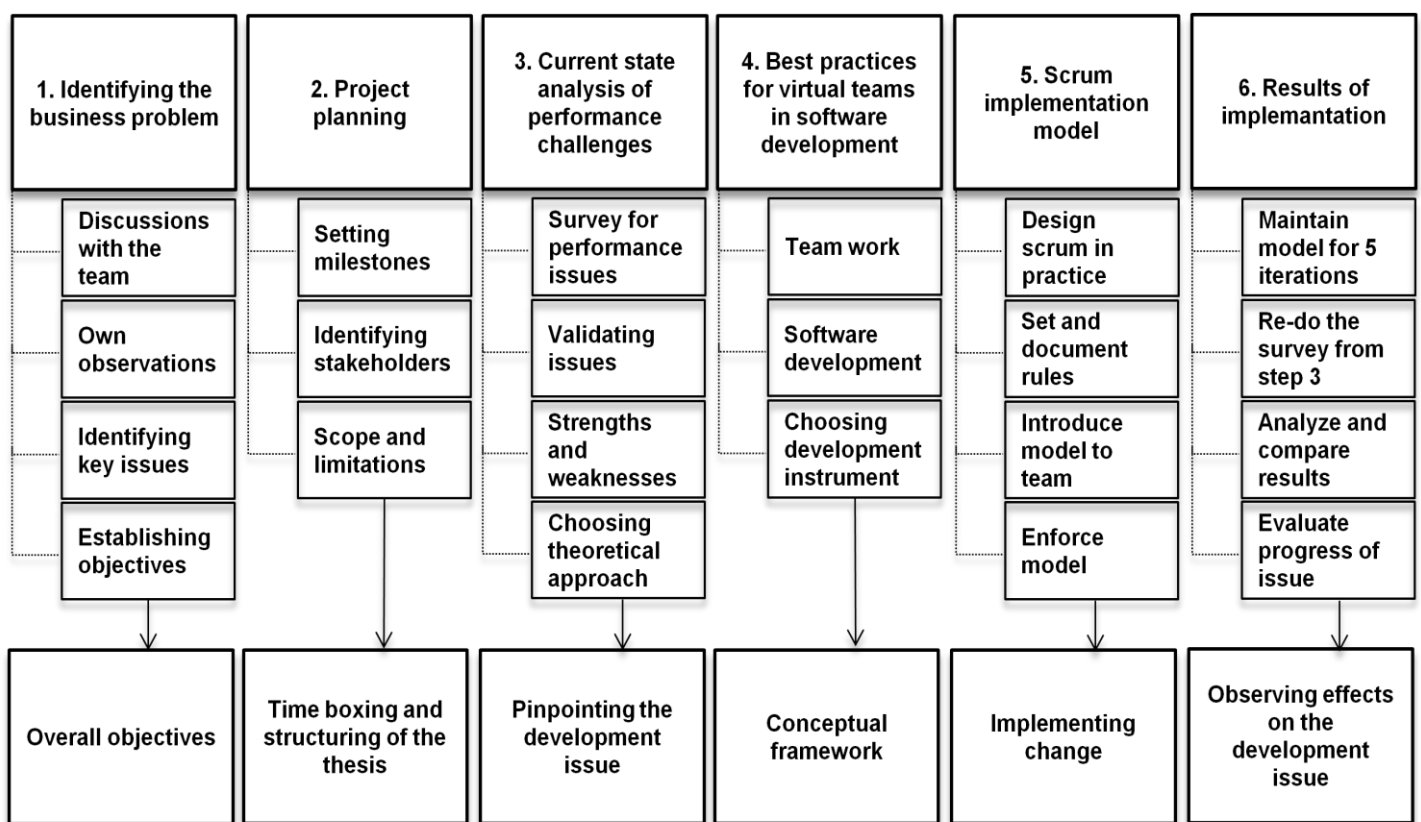


Figure 3. Steps and structure of the thesis

1) Identifying the business problem

In this step the acute business problem is identified and discussed as a part of everyday work. Typical discussions about the subject were vague at this point; team members addressed the issues through their symptoms instead of the actual cause; “this issue was supposed to be ready two weeks ago” or “our customer is very angry because this issue broke down”. As these conversations continued, the tone of

discussion changed to suspicion “what is the team [in Poland] doing?” “We must direct their work more closely”. On the other hand, the Polish team grew frustrated feeling that nothing they did was sufficient; as previously finished issues required repairing but at the same time more issues were being demanded to be built, the atmosphere grew from collaboration to issuing blame on others why the team was not performing. These kinds of problems with virtual teams are quite common; as the team lacks face time it is hard to appreciate other member’s efforts for leaving the office late and arriving early the next morning which is visible in co-located teams (Nunamaker, et al., 2009). Virtual team members have also much more competing demands for their attention in their immediate workplace. (Nunamaker, et al., 2009) After a few months of this discussion, it came apparent that the issues had to be addressed. This became the root for the business problem to be solved; what kind of problems does the team have and how they could be solved? Together these became the key issues and overall objectives for the thesis.

2) Project planning

In this step the progress, preliminary content and timetables of the thesis were planned. The key stakeholders were identified as the development team in Poland and the business and production management team in Finland. The nature of the study was then planned; a case study using mostly quantitative data to identify acute and actual problems and present practical tools to solve them. This step also guided identifying the scope and limitations of the thesis. Outcomes of this step were the time boxing of the data gathering period of the thesis (summer to fall 2014) and of the entire thesis (summer to winter 2014).

3) Current state analysis of performance challenges

As discovered in step 1), the nature of the problems was vague. The purpose of this step was to examine the current situation critically, what was the cause for problems described in step 1 and how that cause could be addressed. The outcome of this step was the pinpointing of the development issue so it could be addressed through the conceptual framework.

4) Best practices for virtual teams in software development

In this step the detailed problem from the last step was positioned in the existing conceptual frameworks of both virtual teams and software development. Virtual teamwork theory, its challenges and best practices are introduced from the context of the business problem as well as possible tools or development instruments that software development methods might have to address the problem. The goal of this step was to build the conceptual framework for both the problem and solution.

5) Scrum implementation model

In this step the development tool chosen as a result from the last step is introduced and taken in to use in practice. In this step, following the literature and existing best practices, it is described how the tool was taken into use and how it was enforced. The purpose of this step was to implement the development instrument into actual work environment.

6) Results of the implementation

In this step we first maintain the development model introduced in the last step for five iterations and then re-do the survey we did in step 3. We take the data from both surveys and analyze them to see if there have been any changes. The end purpose of this step is to present conclusions if implementing the development effort had any effect on the business problem described in step 3.

2.2 Research design

The nature of this research is action research. Mark Easterby-Smith et al state that action research includes the following assumptions;

- Studied phenomenon is continually changing,
- The best way to learn about an organization is through attempting to change it and
- The people most affected by, or involved in implementing these changes should be as involved as possible in the research process itself. (Easterby-Smith, et al., 2008)

As the nature of the business problem is such that the author of the thesis is directly involved in solving it as a part of existing job responsibilities, this design approach corresponds very well to the business problem.

3 Current state analysis

It can be helpful to summarise the key issues arising from an analysis of the business environment and the capabilities of an organization to gain an overall picture of its strategic position. SWOT was created based on surveys, own observations and discussions with the team. SWOT summarises the strengths, weaknesses, opportunities and threats likely to impact on strategy development that arise from such analyses. This can also be useful as a basis against which to generate strategic options and assess future course of action. The aim is to identify the extent of which strengths and weaknesses are relevant to, or capable of dealing with, the changes taking place in the business environment. (Johnson, 2009)

3.1 Survey

A questionnaire was sent to the team where the team was asked to describe, in free form, performance problems that they thought the team was having. Questionnaire is attached as appendix 1. Based on the answers, a more detailed survey was then created. The issues were then categorized to belong in time, quality or operational related areas. Survey is attached as appendix 2. The category areas and survey results are presented below. The sample size was 100%, as all team members answered to it. Approximately 40% of the participants had worked with the project for 1-2 years, and 77% of the team was allocated to the project over 50% of their daily work. 50% of the team was allocated to the project for 81-100% of their daily work. A likert scale was used to process the answers. In order to analyze and compare the results, a weighted average (WAVG) was calculated for each question.

3.1.1 Time related issues

These issues had in common time and deadlines. Did the team think that it takes too long to get issues done and how accurate are the deadlines? In software development time is a poor absolute metric, as finished functions can be either very complex or very simple to create. From the business perspective, the Product Manager does not usually have the technical competence to know how complex in practice the functions they requesting are.

Table 1. Time related issues, June

Time related issues	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	WAVG
It takes too long for an issue to become a working, released feature in general	1	4	2	2		2,6
It is difficult to predict with any amount of certainty when an issue is ready after it has been specified	1	3	1	4		2,9
During the implementation, unforeseen problems usually occur that will delay the finishing of the issue		2	5	2		3
When a deadline for a certain feature is set, it usually fails	1	5	3			2,2

Overall the team did not feel that they had particular problems in the delivery time of issues. The team did, however, feel that it was difficult to predict when an issue would be completely ready after specifying it. Majority of the team was vague about how the actual implementation process worked, as over half of the team said that they did not agree nor disagree with the statement that unforeseen problems usually occur during implementation that will delay the completion of the issue.

3.1.2 Quality related questions

Quality related questions addressed the perceptions of the team on the overall quality; did the team feel that when issues are done, they function as planned and do not have bugs and how good the quality of the released issues are in general. Quality is a challenging aspect to measure in software development, as it can be seen as being subjective. A developer may think that good quality in coding means that it does what it supposed to do in the most efficient way possible, with as few a line of code as possible. This might be felt as having intrinsic value, meaning that it does not matter how much time it takes, as long as the issue being done is done “well”. Product Manager may think that good quality means that the issue is done as they specified in as short a time as possible. Customer might think that quality means that the function does what it’s supposed to do and won’t break down.

Table 2. Quality related issues, June

Quality related issues	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	WAVG
After an issue is done and released to production often something that previously worked, breaks down.	1	6	1	1		2,2
Often even after smoke tests, internal revision and quality assurance, the issue might still not work as planned or is broken	2	6	1			2,2
It is difficult to measure the quality of the released issue	1	4	3	1		2,4
If I am the owner of the issue, it usually correspond well to my specifications once it is returned to me as done	1	1	2	5		3,2
I need more information about the overall business strategy and client needs to do my part properly	2	2	2	2	1	2,8
If I need help, I know who to ask and I get the information I need to proceed	1			3	5	4,7
I understand the overall development workflow well	1			6	2	3,8
I understand what is expected from me as a part of the process	1			5	3	3,7

The team did not feel that there were major quality problems; majority felt that issues did work well after they were released and no regression problems occurred. Some dispersion was felt on needing to know more about the overall business strategy and

needs of the clients to be able to perform better. Almost all felt that when they needed help, they knew where to get it from and understood the overall development workflow well.

3.1.3 Operative issues

These issues were about the development cycle in general. Did the team feel that they knew what was expected from them, did they feel that they had the necessary information about relevant areas to do their jobs successfully and did the team recognize internal problems in the development cycle, as erratic prioritization from the Product Manager.

Table 3. Operative issues, June

Operative issues	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	WAVG
Issue prioritization is erratic; it changes so often that it has a negative impact in my work	2	4	1	2		2,3
Reasoning behind changing prioritization is not clear enough	1	7	1			2,1
Reasoning behind the issues (why something is done) is so unclear that it has negative impact in my work	1	7		1		2
Once I get an issue assigned to me, I understand what is needed from me			7	2		3,2
Once an issue is rejected and returned to me, I understand clearly why			2	5	2	4
At this point of the product's lifecycle, I feel it is more important to release new features fast	1	4	3	1		2,4
At this point of the product's lifecycle, I feel it is more important to release fewer features, and use more time to make sure that the features work well			6	3		4,3

Notable findings were that a majority of the team were vague when asked if they understood what was needed from them once an issue was assigned to them. If issues were rejected and returned to them to do again, the team understood well why. Few team members thought that the issue prioritization was erratic.

3.2 SWOT

Based on the survey, own observations and discussions with the team and stakeholders, the following SWOT of the current state were created. The findings were then used to help identify how valid the chosen development method would be to address the weaknesses in a virtual team context, which is shown in table 6. Even though the opportunity to change nearshoring partner, change the team as a co-located team or even divesting the whole service were recognized, these were not considered further in this setting as the cost of change would become too great.

Table 4. SWOT

Strengths	Weaknesses
<ul style="list-style-type: none"> ○ Experienced development team ○ Good knowledge of different software development methods ○ Free to use whatever development method available ○ Free to change practices that don't work 	<ul style="list-style-type: none"> ○ Information gap between business and development ○ Predictability on when issues are ready ○ Vulnerable to switching priorities ○ No possibility for continuous development
Opportunities	Threats
<ul style="list-style-type: none"> ○ Change near shoring partner ○ Change near shoring to co-located team ○ Divest whole service and substitute it with a new one 	<ul style="list-style-type: none"> ○ Competition uses co-located teams and have competition advantage with a faster development cycle ○ Breaks in data networks and Internet ○ Political situation in Eastern European countries

4 Conceptual framework

In this chapter the conceptual framework for the thesis is presented. As the team as a whole is separated in two different countries, languages and culture, it is beneficial to explore what teams are and what kind of characteristics and challenges does virtual teams have. In order to decide the most suitable tool to use in practice, agile and traditional software development methods are introduced as well as why the current Kanban method was not the best possible solution for the team.

4.1 Group or a team?

There has been a lot of discussion about the difference between groups and teams in the literature. In their book *Making Sense of Change Management*, Esther Cameron and & Mike Green present these following findings;

Schein and Bennis (1965) suggest that a group is ‘any number of people who interact with each other and are psychologically aware of each other and who perceive themselves to be a group’. Morgan *et al* (1986) suggest that ‘a team is a distinguishable set of two or more individuals who interact interdependently and adaptively to achieve specified, shared and valued objectives’. Sundstrom *et al* (1990) define the work team as ‘a small group of individuals who share responsibility for outcomes for their organizations.’ (Cameron, 2012)

Cameron and Green conclude;

A group is a collection of individuals who draw a boundary around themselves. Or perhaps we from the outside might draw a boundary around them and thus define them as a group. A team on the other hand, with its common purpose, is generally tighter and clearer about what it is and what its *raison d’être* is. Its members know exactly who is involved and what their goal is. (Cameron, 2012)

As the studied group of individuals explicitly have a shared responsibility of their collaborative outcomes and their reason for existing is clear, it can be said that in the context of this thesis the proper perspective is “a team” instead of “a group”. In the studied environment the team can be seen as a part of a group, where the group consists of the whole business unit. Under the business unit there are several services, or teams. Employees are then either dedicated to a specific team, or they might be a

shared asset who does work for other services as well. An example for the first is a development team which maintains and develops a single product. An example of the latter could be marketing or sales, who do work for all services. Team/group dynamics and the role of individual assets are shown in the following figure

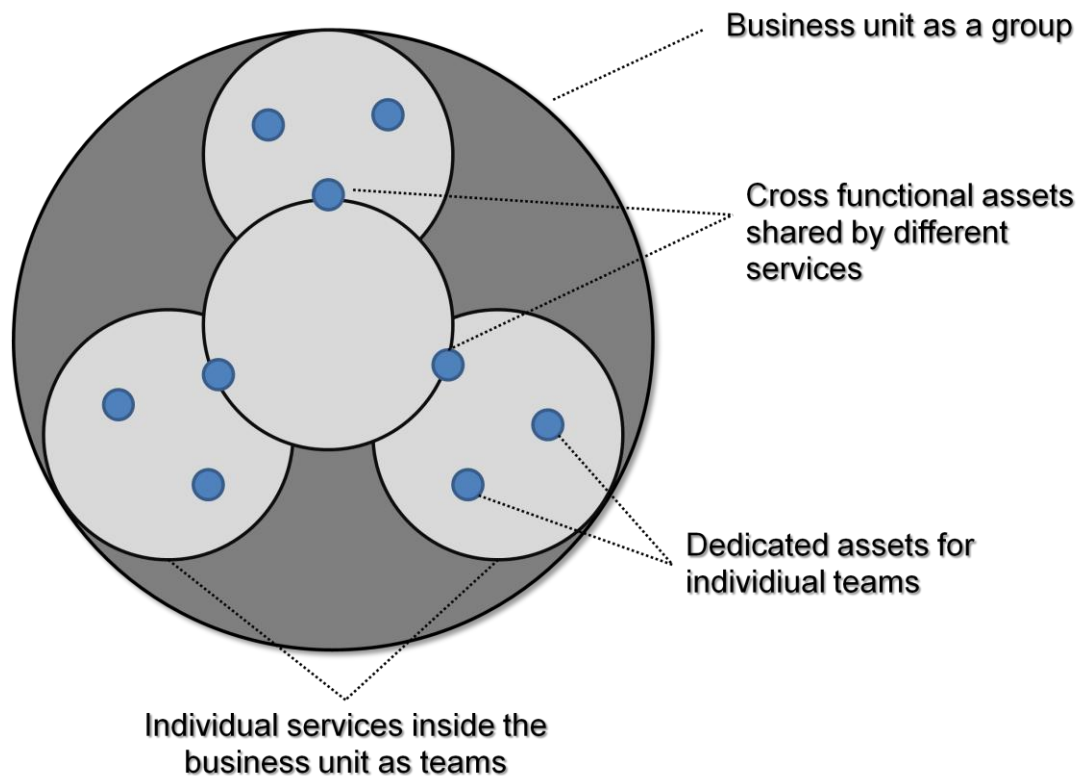


Figure 4. Teams within a group

4.2 Virtual teams

Virtual teams are teams whose members are not located physically in the same location and either never meet or meet only rarely. (Cameron, 2012), (MacDonnel, et al., 2009) Following the globalization of the business environment and advancements in technology enabling high quality communication methods, virtual teams are becoming more commonplace. (MacDonnel, et al., 2009) As projects become more and more complex, they also require a very diversely talented multi-disciplinary team to perform them (Tuffely, 2012). Many government and military organizations are forced to scatter their organization because of terrorist threats or natural disasters. Cutting costs by outsourcing support functions to other locations and countries is also becoming more and more common (Nunamaker, et al., 2009).

4.3 Challenges in virtual teams

Common challenges in virtual teamwork are for example establishing trust between team members. This together with a feeling of isolation by using only virtual communications methods is thought to decrease team cohesion (MacDonnel, et al., 2009). It is also noted that the level of trust has a strong positive relationship with team performance as well as team work satisfaction. (Hungwei & Heng-Yu, 2011) Those teams that know they can rely on each other to get everything done enjoy a higher level of team cohesion and are more efficient (Panteli & Tucker, 2009). Besides trust, one major factor can be the actual methods of daily interaction. In co-located teams it is easy to go and discuss issues as they come along, but this is not so easy to do in virtual environment. Virtual teams often do not have the methods which come as natural to co-located teams to establish and maintain a shared understanding about the nature of their tasks. (Nunamaker, et al., 2009), (Panteli & Tucker, 2009) This makes expressing the desired outcomes, expectations and ways of work more important in virtual teams than in regular teams. (Nunamaker, et al., 2009). In virtual environment, a good strategy is to lead by subtle influence and allow team members exercise their sense of self-government and empower them with influence (Tuffely, 2012). If possible, the team should be free to choose the tools they want to use to get the job done, and also decide among themselves how the work is organized. (Library Technology Reports, 2008) Successful teams are also those who support other team members to do great work (Library Technology Reports, 2008). These issues have two different conclusion points that can be seen to require special attention between virtual and regular teams. First, the overall goal setting of what needs to be done must be clearly communicated and ensured that it is understood in the same way; studies suggest that in the teams that work well, shared goals have been used to create a higher level goal, which was focused on the success of the team as a whole. (Panteli & Tucker, 2009) Second, a virtual team needs to be facilitated in a way that is tailored for their environment. (Library Technology Reports, 2008) Overview of virtual team challenges are presented in the following figure.

Table 5. Challenges facing virtual teamwork (Nunamaker, et al., 2009)

Challenges Facing Virtual Teamwork

• Loss of non-verbal cues
• Reduced mechanisms for informal conversation
• Reduced opportunities to build friendships
• Time zone differences
• Complicated, unreliable technology
• Building consensus at a distance
• Establishing shared meaning at a distance
• Different work processes
• Different cultures

4.4 Enabling trust

It is not enough that a team consists of individual who have the right skills. The group must also work well together. Trust is often something that is missing when professionals work together. This might not be a serious issue if the work is routine and their business impact is small. If the team has a bigger responsibility, the team needs to have trust (Kotter, 2002). Some people might avoid teamwork because of it means that their individual performance is seen in the context of the team. Individuals might think that even if they do their own work well, but the “others” do not, they will be treated unfairly. (Katzenbach, 2002). When team has a high degree of trust and feel themselves truly committed, they confront each other without fearing defensiveness or retaliation. In this setting they are helping someone to get back on track on issues that are not going well (Lencioni, 2012). Condensing the literature findings about team work and their challenges, a simple evolutionary progress cycle can be identified. Virtual teams thrive on trust, and trust can be instilled for each team member by first communicating and agreeing very explicitly on the way things should be done so everyone knows what is expected from them. The higher of the autonomy of the team is for choosing the way of doing things, the better the results are. This is a similar

progress than in the so called Deming's quality circle, where continuous development is achieved by following a Plan-Do-Check-Act progress (Deming, 1986).

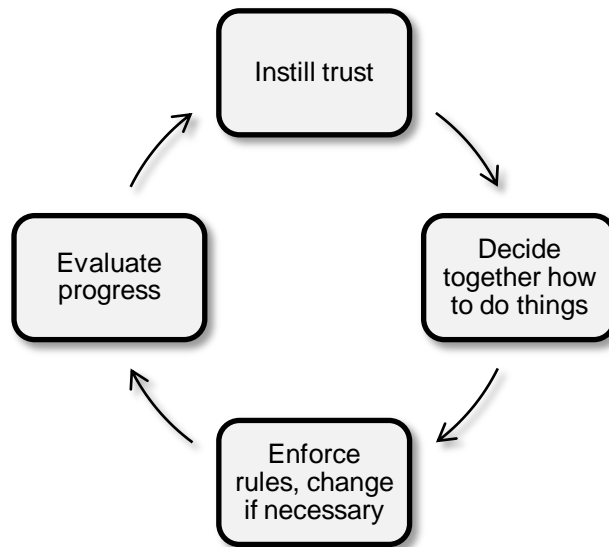


Figure 5. Evolutionary progress of creating trust in virtual team environment

4.5 Agile and traditional software development methods

Software development methods have progressed vastly during the last decades. Today they can be divided in two fundamentally different approaches, agile and traditional. In the following, agile and traditional software development methods are introduced.

4.5.1 Traditional software development

The so called traditional software development takes a plan-driven view which assumes that problems are very well defined beforehand and the results of the actual development action are well predicted. This creates extensive up-front planning so that the entire development effort can be predicted beforehand. (Devi, 2013) Very common traditional software development model was the waterfall.

4.5.1.1 Waterfall

Traditional methods of software implementation followed the so called waterfall method, a plan-driven process created first in 1970 by Winston Royce. It describes a

hierarchical, step-to-step process on how to plan and implement software development. It was later introduced to the software industry in general use.

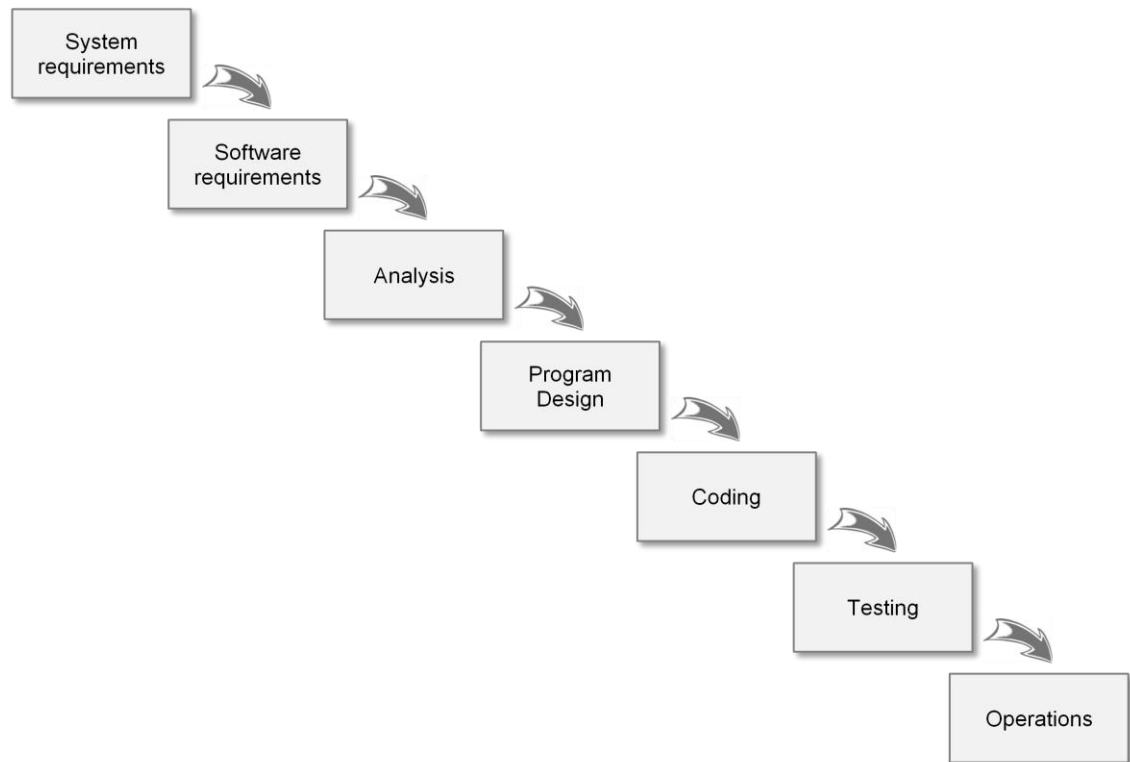


Figure 6. Waterfall development (Royce, 1970)

This method has received a lot of criticism in the last decade, mainly due to the time and effort it takes to create the first actual and functioning feature. So long, that the actual need for the function could have been changed many times since the first “software requirement” step of the process was completed. The model requires that the previous step is always complete before progressing to the next one. In software projects, this almost never achieved (Dooley, 2011) (Schwaber & Beedle, 2002).

4.5.2 Agile development

Agile software development is a response from the software industry to service businesses in a fast paced and global setting. It has its roots in the agile manifesto, a set of four principles created by a group of seventeen independent software development experts in a summit in 2001. (Alliance, 2014) These principles are;

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

(Beck, et al., 2001)

4.5.2.1 Kanban

Kanban is a production control method introduced in the car production industry during 1970's (Y. Sugimori, 2007). After its successful introduction in automotive industry, it has been recognized as useful system for other uses too, for example software development. (Shore Labs, 2014) There are 5 core properties to a Kanban implementation;

- 1) Limit work-in-progress. This means that in order to optimize the passing of the issues between different steps, there must be a limit on issues existing at the same time in each step. By adhering to the limitations, it is easy to point out problems in the work flow to resolve them. (Anderson, 2010)
- 2) Visualize the workflow. This helps the stakeholders to understand what issues are waiting to be done, what are being done and what has been done. Visualizing the workflow can be done for example by using card walls with cars and columns, where cards represent the issues being done, and columns the different steps in the workflow. (Anderson, 2010)
- 3) Managing the flow. Here the flow is analyzed through the first two steps, recognizing problems and solving them and repeating the cycle to see if the change had a positive or a negative effect on the desired issue. (Anderson, 2010)
- 4) Make the process explicit. This means that all team members need to have a clear and shared understanding on how the process works and make sure everybody follows the same rules. This makes the discussion more rational, empirical and objective. (Anderson, 2010)
- 5) Improve collaboratively. By following the first four steps, it is possible for the team to build a shared comprehension of the problem and suggest improvement actions which can be then agreed by consensus. This is also known as "Kaizen", meaning of continuous improvement. (Anderson, 2010)

After analyzing the current state and context of the problem, it was seen that the current Kanban method might not be optimal for this team. Kanban was not being implemented successfully. Kanban implementation lacked the last three steps that were essential to make it work; the work flow was not analyzed systematically to pinpoint problems, process was not being made explicit to all team members which made the last step, continuous development, impossible. Instead of trying to improve the Kanban method an alternative development methods was introduced; Scrum. Scrum is an agile framework for creating complex projects. (Scrum Alliance, 2014). One advantage of Scrum over Kanban is the possibility to set and predict certain amount of issues to be done in an agreed timeslot. This was seen as a desirable development, which Kanban could not provide. This was one main reason for not trying to improve the current method and instead change the method altogether. In table 6, it is shown how the strengths of Scrum can answer to the challenges in virtual teams.

4.5.2.2 Scrum

Scrum is the most used agile product development framework. Scrum Alliance, a non-profit membership organization that encourages and supports the widespread adoption and effective practice of Scrum (Scrum Alliance, 2014), describes Scrum like this;

The concept that would become Scrum was first introduced to the world in 1986 by Hirotaka Takeuchi and Ikujiro Nonaka in the "New New Product Development Game" (Harvard Business Review, January/February 1986). They defined their approach as a "flexible, holistic product development strategy" and proposed it would result in fast, flexible product development. They called it the holistic or "rugby" approach because, much like in a rugby match, one cross-functional team passes the "ball" back and forth on the way to the "goal line." This was, and continues to be, in stark contrast to approaches that progress in a rigid, linear fashion. (Scrum Alliance, 2014)

Ken Schwaber and Mike Beetle, two members of the original Agile Manifesto describe Scrum further;

Scrum is a management and control process that cuts through complexity to focus on building software that meets business needs. Scrum is superimposed on top of and wraps existing engineering practices, development methods and standards. Scrum deals primarily at the level of the team. It enables people to work together effectively, and by doing so, it enables them to produce complex, sophisticated products. Scrum is a kind of social engineering aiming to achieve the fulfilment of all involved by fostering cooperation. Using Scrum, teams

develop products incrementally and empirically. Teams are guided by their knowledge and experience, rather than by formally defined project plans. In almost every instance where Scrum has been applied, exponential productivity gains have been realized. (Schwaber & Beedle, 2002)

Scrum is iterative, continuous process of increasing performance in almost all areas of team work. Here we can see a similar link to the Deming's quality circle, presented earlier in association with the evolutionary progress of creating trust in a virtual team environment. Scrum also enables the continuous improvement, Kaizen, introduced earlier when discussing Kanban.

Particularly interesting for this thesis is the notion of "social engineering". Most studies involving Scrum concentrate on the technical implementation and does not discuss so much the social aspect involved. In this sense, Scrum can be seen as a team work enabler too, instead of just a software development method.

4.5.2.3 Scrum team

Scrum team does not follow titles or functions; it is by nature cross functional, including people with the necessary skills to meet what is needed from the team. (Schwaber & Beedle, 2002) Optimal Scrum self-organizes so that everyone contributes to the outcome. Usually a Scrum team includes at least one senior software engineer and one or more junior software engineer. Quality assurance testers, technical writers, UX designers, graphical artists and other individuals with particular skill sets might be included in a team. Team might also include individuals who are called only to address particular issues, related to technical or business aspects to which the team needs help during the sprint.

4.5.2.4 Scrum master

Scrum master is responsible for the success of Scrum. He or she is responsible that Scrum values, practises and rules are enacted and enforced. Scrum master represents management to the team and team to the management. Scrum master listens closely the daily work the team is doing and does everything in his or hers powers to ensure that the team can work in peace and minimizes disruptions from individuals that are not a part of the team. (Schwaber & Beedle, 2002)

4.5.2.5 Product Owner

Product Owner represents the voice of the end customer, decides what the team does and in which order, manages the backlog, answers functionality related questions from the team and overall is in charge of the product strategy and how it should work.

Product Owner and the Scrum Master must have a very good interaction and common understanding to be able to execute Scrum successfully. (Pichler, 2007)

4.5.2.6 Product backlog

The product backlog is a list of all work that is planned for the team to do at a given moment of time. Product owner is responsible of what is on the list and describes them to the team when implementing them. No issues are allowed to come to the team unless they are in the backlog. Product owner also decides the priority of the items so that the issues that are being taken into development next are always at the top. (Schwaber & Beedle, 2002) (Pichler, 2010)

4.5.2.7 Sprint planning

In sprint planning, the team meets first with the product owner, management and other relevant stakeholders to figure out what is needed to be built by the team during the next sprint. Here it is important to note that the agreed work must be something that the team thinks is possible to do in a fixed time frame. This is a good method for the non technical members of the team to gain understanding on how big effort is needed for the functionalities wanted. After this meeting, the team agrees by themselves how the required functionalities will be done. After planning, the team is committed to do the issues agreed in the planning. By default, the team is free to choose how the agreed work is done. Backlog is also updated, depending on the outcome of the planning. (Schwaber & Beedle, 2002) (Pichler, 2010)

4.5.2.8 Sprint

A sprint is a previously agreed upon duration for delivering a product increment. Ranging from one to four weeks, average being two weeks, it is a concrete timebox within which all issues agreed on sprint planning must be done. If the team discovers that they have selected too much to be done, the Scrum master meets with the product

owner to identify and agree what can be removed from the selected sprint. (Schwaber & Beedle, 2002)

4.5.2.9 Daily Scrum

Daily Scrum is a meeting where the team meets every day for about 15 minutes to explain what it has accomplished since the last meeting, what it is going to do before the next meeting, and what obstacles are in its way. This makes it easy for the Scrum Master to see if team members are struggling or performing well in their daily tasks to accomplish the previously agreed tasks. (Schwaber & Beedle, 2002)

4.5.2.10 Sprint review

Each sprint ends with a sprint review or a demo. Here, the team presents their work for the product owner who was responsible for the backlog as well as other interested parties. The work of the previous sprint is either accepted or rejected. It is recommended for the team members to present their own work, meaning that the team member responsible for certain functionality will be the one showing how it works. If the sprint is accepted, the team will start the iteration again with sprint planning. If issues from the sprint are rejected, they are introduced in the next sprint backlog if the product owner so chooses. (Schwaber & Beedle, 2002).

4.5.2.11 Sprint retrospective

After the review, the team gets together usually without the product owner to discuss what went well in the last sprint, what did not go well and what to change to improve the next sprint. (Mitch Lacey & Associates, 2014) Overview of the sprint process is shown in the following picture.

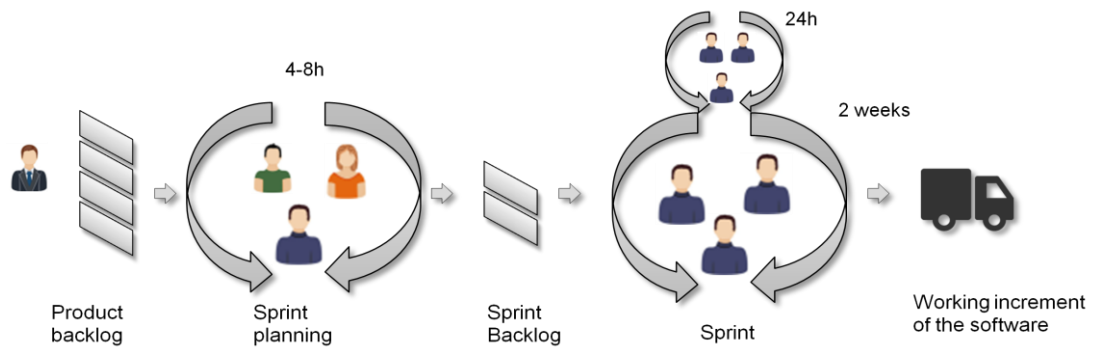


Figure 8. Scrum workflow (Scrum Alliance, 2014) and (Schwaber & Beedle, 2002)

4.6 Solving virtual team challenges and SWOT weaknesses with Scrum

To see how Scrum could be used to address both the problems identified in the SWOT (table 4) and challenges in virtual teams (table 5) a following comparison can be created. Challenges and weaknesses are collected on the left and on the right it is shown how Scrum can help solve them.

Table 6. Solving virtual team challenges (Nunamaker, et al., 2009) and SWOT weaknesses with Scrum

Challenges Facing Virtual Teamwork and recognized weaknesses from SWOT	Solutions with Scrum
<ul style="list-style-type: none"> Loss of non-verbal cues 	<ul style="list-style-type: none"> Lower the significance of non-verbal cues by using instant messaging, pre-formulated templates for issues and documentation. High quality video conferencing
<ul style="list-style-type: none"> Reduced mechanisms for informal conversation 	<ul style="list-style-type: none"> Promote usage of instant messaging in both formal and informal issues when possible. Follow Scrum rules and allow also informal discussions to a degree in weekly meetings.
<ul style="list-style-type: none"> Reduced opportunities to build friendships 	<ul style="list-style-type: none"> Discuss in detail the roles of all members in Scrum. Promote professional trust and friendship.
<ul style="list-style-type: none"> Time zone differences 	<ul style="list-style-type: none"> Decide mutually what time zone is followed for shared meetings. Do not dictate virtual teams schedule in other ways
<ul style="list-style-type: none"> Complicated, unreliable technology 	<ul style="list-style-type: none"> Choose the best working tool for the need. If the tool blocks the daily work, flag it to upper management immediately and demand change
<ul style="list-style-type: none"> Building consensus at a distance Predictability on when issues are ready 	<ul style="list-style-type: none"> Sprint planning creates consensus as the team decides together what will be done and how complex it is estimated to be
<ul style="list-style-type: none"> Establishing shared meaning at a distance Vulnerable to switching priorities 	<ul style="list-style-type: none"> Sprint planning, demos and backlog grooming helps making sure what is wanted from the product and from the team
<ul style="list-style-type: none"> Different work processes Information gap between business and development 	<ul style="list-style-type: none"> Follow Scrum. Make the rules explicit and make absolutely sure that all team members know what they are. Let the team decide as much as possible considering on what tools for example, are used.
<ul style="list-style-type: none"> Different cultures No possibility for continuous development 	<ul style="list-style-type: none"> Use Scrum to overcome cultural challenges; scrum can be new to both parties which then generates a new, but shared way of values and methods of doing work

5 Case study application

In order to address the expressed problems within the development cycle, Scrum workflow management was decided to be implemented instead of the existing Kanban model. Affected stakeholders were identified to be the development team in Poland, Production Manager and Product Manager in Finland. When needed, other stakeholders were invited to contribute for completing different issues. (Figure 9)

From the start, it was decided that the Scrum model would be as strictly followed as possible. However as the team existed in two different countries, all aspects of the model was not feasible to enforce, such as the daily Scrum. It was also decided that in order to evaluate the effectiveness of the new model, the same survey that was used to recognize the problems within the development cycle would be re-done after a few iterations of the Scrum model, in order to establish if it had any effect on the issues that were seen problematic. The Production Manager in Finland was responsible in enforcing that the day to day operations followed Scrum.

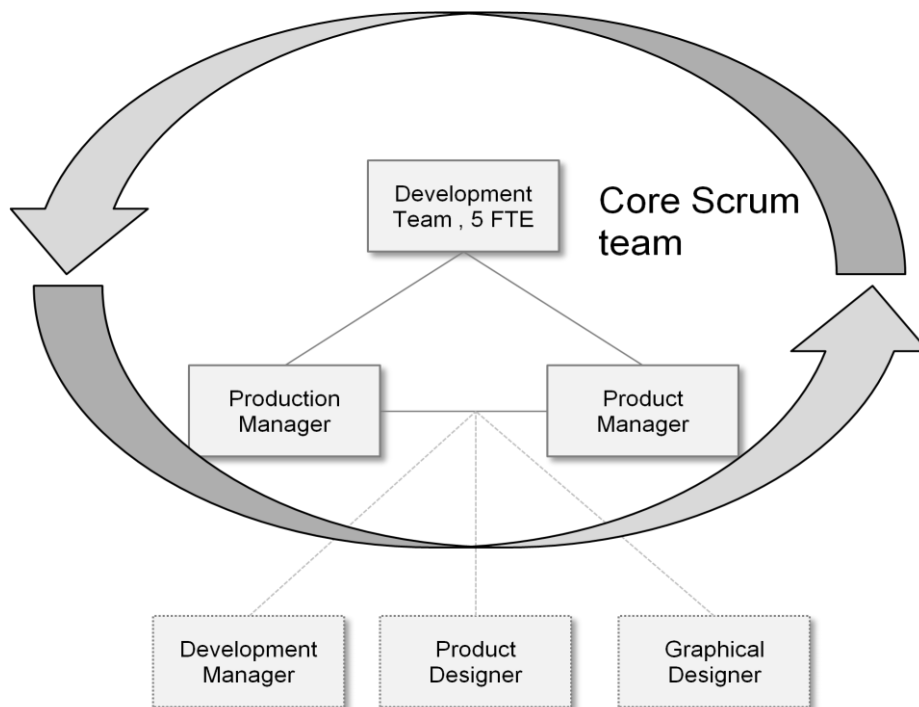


Figure 9. Scrum team and other stakeholders

5.1 Implementing Scrum

To implement Scrum, the Finnish Production Manager together with the Development Manager planned how the new method would be taken into use. The team already had experience working with Scrum, so the method was not entirely new. Biggest change was to introduce the Finnish Product Manager more closely to the daily work of the development team. In sprint planning, the Product Manager would discuss the issues with the team and provide clarifications on the issues that were taken into the sprint. Production Manager was in charge of configuring the used tools to support Scrum and to enforce that the daily and weekly development cycle was followed. To support Scrum, existing project management and issue tracking tools were used. Atlassian Jira is one of the most common issue tracking and project management tool in the world. It is used by 25,000 customers in over 122 countries. Jira provides tools to support both Kanban and Scrum workflow. Atlassian Confluence was used to document processes and larger functionalities. (Atlassian, 2014)

5.2 Scrum calendar and workflow

Scrum calendar was drawn and visualized to the team. This calendar showed everyone what happened during each day of the week, and when the different parts of Scrum were done. Sprint length was decided to be two weeks, except during the summer holidays when it was longer. As the team worked in two different time zones, schedules were agreed to follow the Eastern European Time.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
(EET)			New sprint starts									
9:00												
10:00	Daily	Demo	Sprint planning pt. 1	Daily	Daily			Daily	Daily		Daily	Daily
11:00			Sprint planning pt. 1							Backlog grooming		
12:00			Sprint planning pt. 2									
13:00			Sprint planning pt. 2									
14:00		Retrospective										
15:00												
16:00												
17:00												

Figure 10. Scrum calendar

Different work flow steps were documented. All issues must had to pass these steps during the actual development sprint.

Workflow

Ready for development:

Issues that are included in current sprint. Work of this issues hasn't started yet. This pile should be ordered by priority.

In progress:

Work started. Developer is working with issues in development environment.

Internal revision:

Task is finished and it's reviewed by another developer.

Quality assurance:

Issues is in acceptance testing assigned to PO.

Ready for deployment

Issue is tested and accepted and it meets the definition of done (**todo**)

Figure 11. Workflow steps

5.3 Backlog

Backlog was created to hold all issues that were planned for the team to do. Product Manager was responsible for the prioritization of the issues in the backlog; the team would always start sprint planning from the top of the issues in the backlog. It was important to make sure the sanctity of the backlog was enforced; everything the team did was required to be in the backlog. Between sprints, backlog grooming took place between the Product Manager and Production Manager, to ensure that the issues were mature enough for the team to discuss in the next sprint planning.

Issue ID	Description	Label	Points
AMAIN-1595	Allow access to stage environment only for XXXXXXXXXX		
AMAIN-1574	First name, lastname should be optional when creating a profile	Profile API	2
AMAIN-1586	Functionality to update modified time in profile	Profile API	2
AMAIN-1587	Refactoring / finetuning API	Profile API	2
AMAIN-1590	Field restrictions in profile API	Profile API	
AMAIN-1591	profile ids in recommendations	Profile API	
AMAIN-1581	Planning - Search from attachments		
AMAIN-1582	Planning - XXXXXXXXXX		
AMAIN-1214	Taito statistics		5
AMAIN-1569	When creating a recruitment from proposal, I want to import the XXXXXXXXXX	Proposal improvem...	8
AMAIN-1509	Create default job ad, application form and publication when creating a company		2
AMAIN-1562	clean XXXXXXXXXX server		5
AMAIN-1512	Profile attachments, deleting and then adding new attachments not working		2
AMAIN-1301	Upgrade XXXXXXXXXX to newest version (2.5)		
AMAIN-1269	As a recruiter I want to see description and name job job ad when selecting it to public		1
AMAIN-494	Empty the notes textarea field after save		0.5
AMAIN-998	The user list in recruitment user management should be ordered by name desc		1
AMAIN-1264	Company Styles/Job ad Images - Chosen image doesn't show in the field		2
AMAIN-693	Preview of job ad image in company styles causes http 404		2
AMAIN-1530	Taito should not log out recruiter in middle of use when the recruiter looks publications		2
AMAIN-708	User reports not working for some customers		

Figure 12. Backlog

5.4 Scrum board

Scrum board was created to visually assign issues from the backlog for individual developers. Scrum board also included different phases that individual issues must pass during a sprint.

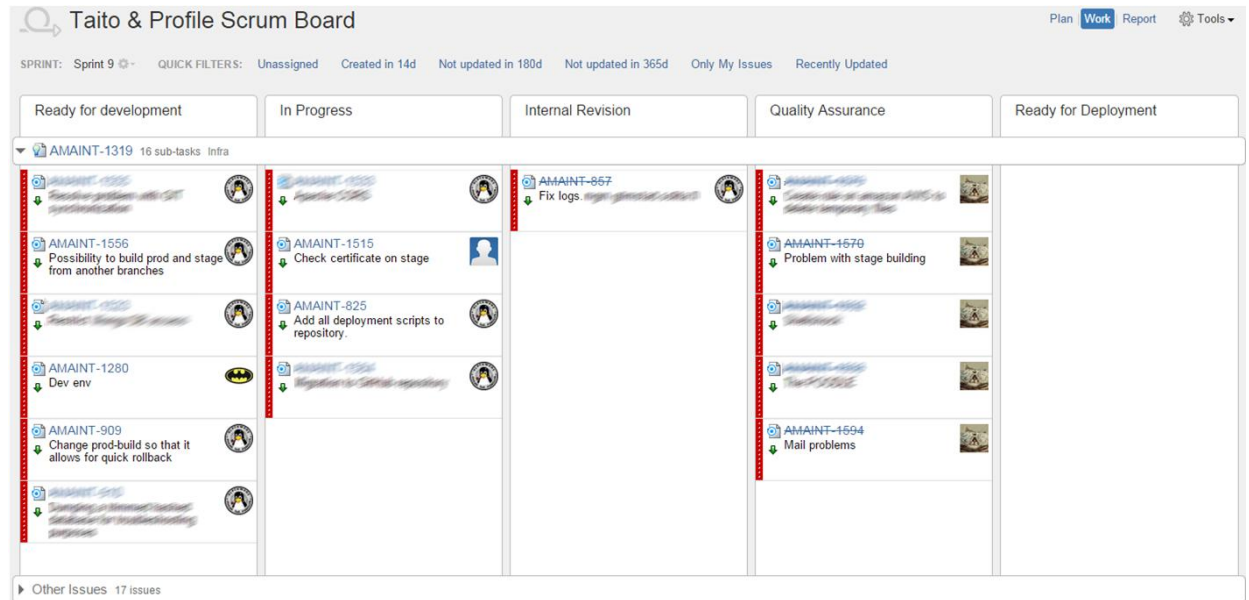


Figure 13. Scrum board

5.5 Sprint reports

As the issues were tracked in Jira during a sprint, it was possible to visualize what issues were chosen for the sprint and how they were achieved during the sprint. The used tools provided different reports, such as burn down chart and team velocity. The burn down chart visualized the relation between the time left in a sprint versus the work still left.

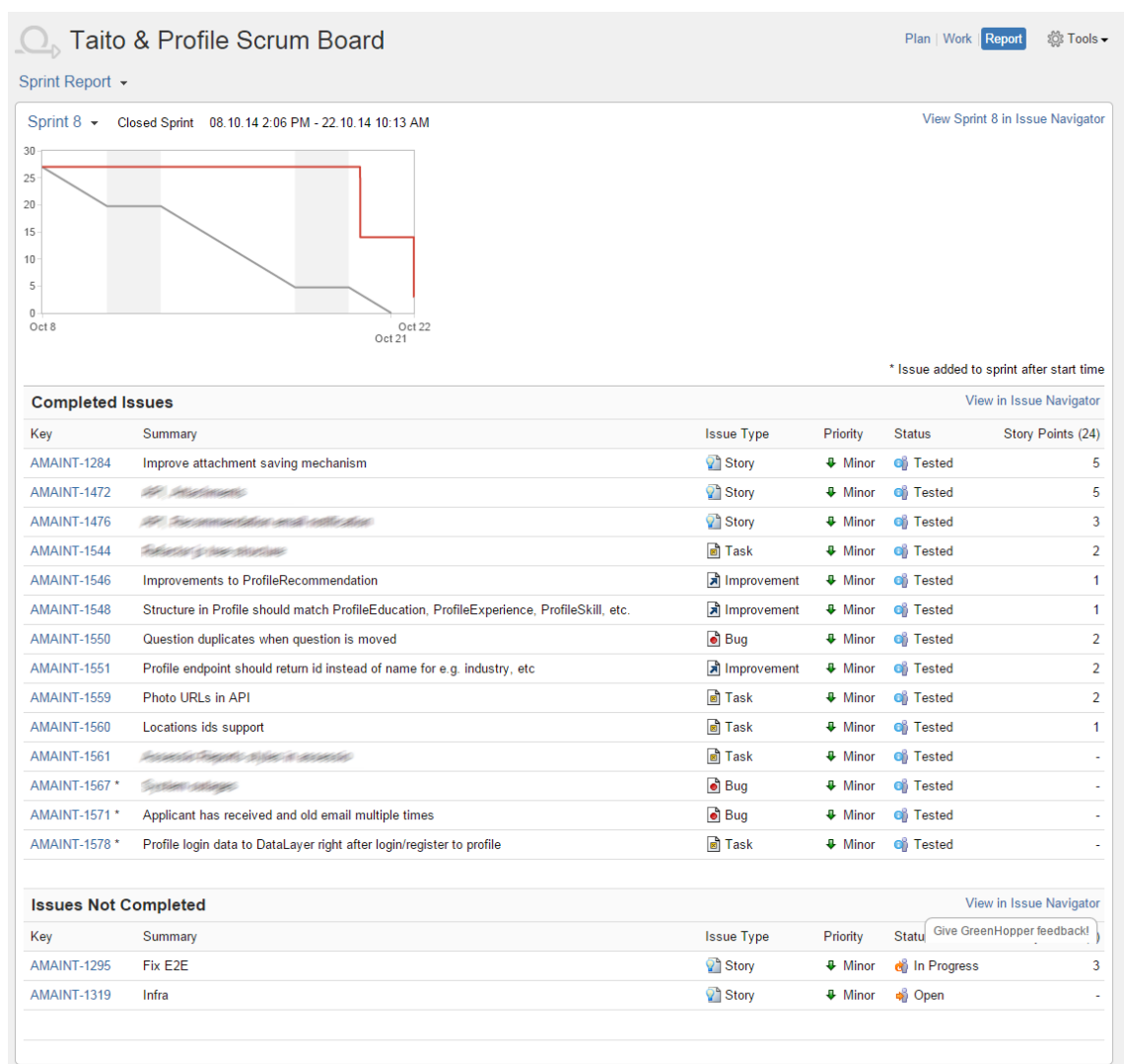


Figure 14. Burn down chart and completed issues in a sprint

On top of burn down charts, it was possible to calculate the team velocity. Velocity is a metric used in software development to give insights to how complex issues have been done, when the team is able to deliver issues from the backlog and how complex issues can be delivered by a certain date (Oliveira, 2014). After each sprint, the team will become more efficient at judging how complex the issues are from the backlog which creates calculable possibilities to predict with a certain probability on when a certain feature will be done.

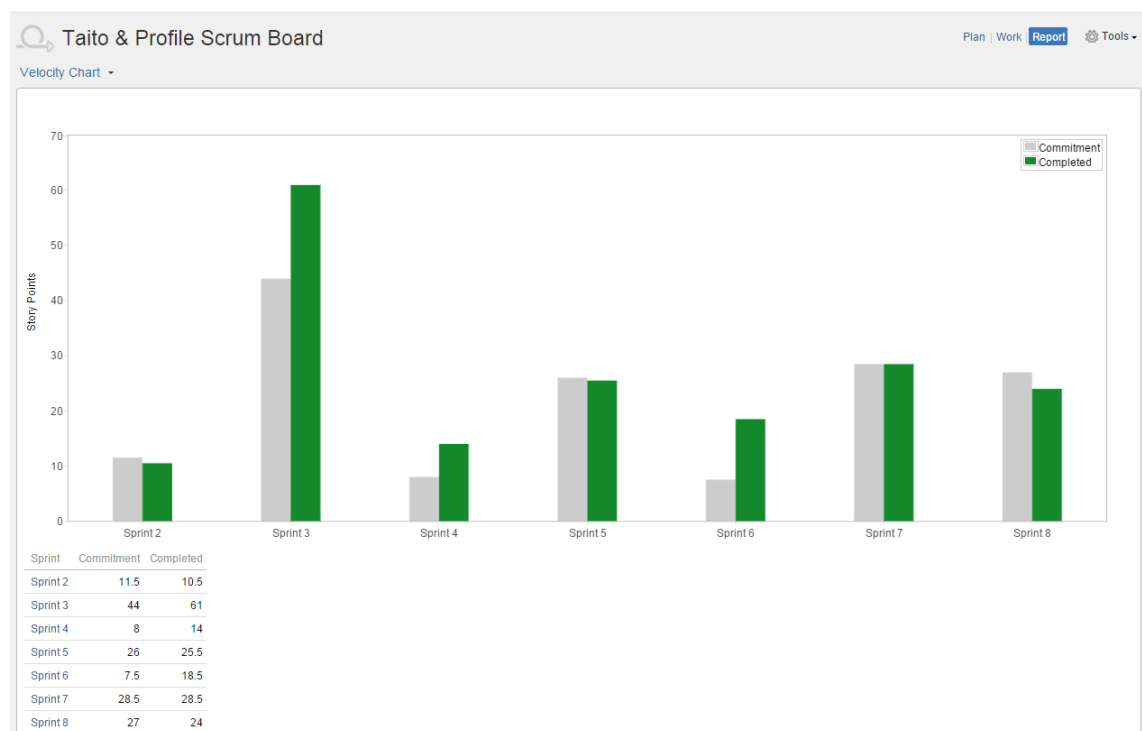


Figure 15. Team velocity

5.6 Sprint planning

Every two weeks, a new sprint was started after a two-part sprint planning session. In the first part, the Finnish and Polish teams met via high-quality videoconference where the backlog was shared in real time and the items to be included in the sprint were chosen according to their priority. After this, the development team in Poland discussed the issues in more detail and presented more detailed questions about the issues if needed through instant messaging tools. When everything was clear, the sprint was started. Sprint planning session is presented in the picture below.

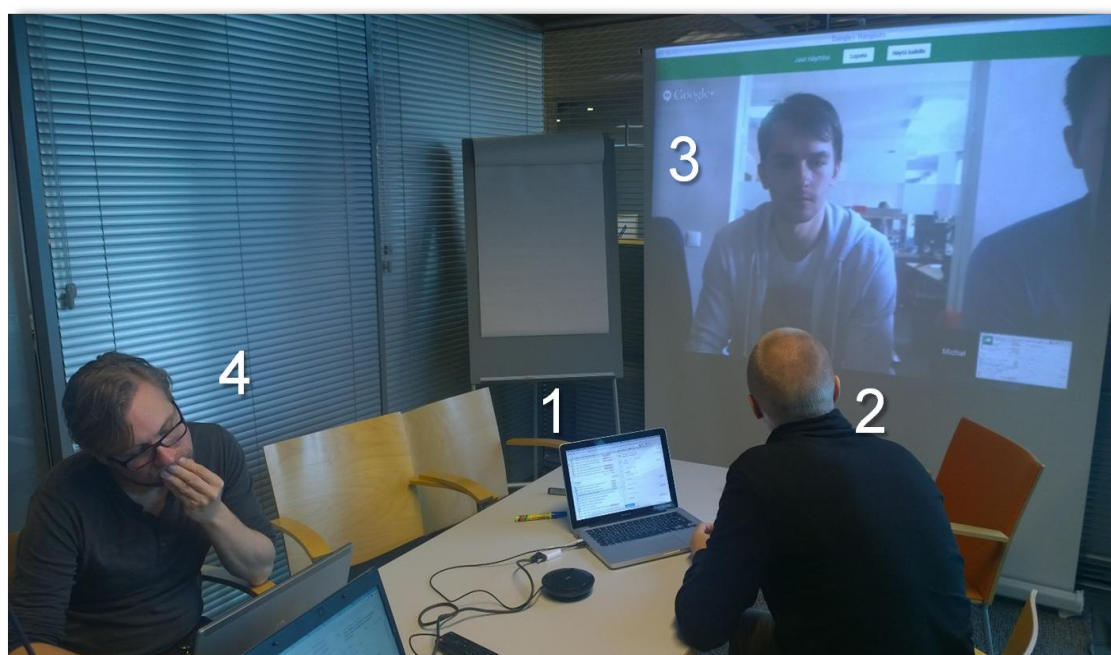


Figure 16. Sprint planning in progress

The numbered items in the picture are;

1. Items from the backlog are chosen to the sprint. Backlog is visible in real time for all of the team
2. Production Manager discusses with the development team
3. The team asks questions and comments how complex issues are in the backlog
4. Product designer comments on issues when required

5.7 Instant messaging

Instant messaging was used to discuss on-going issues and solve problematic situations. Instant messaging provided a way to enable informal discussions as well. Instant messaging clients exist both for desktop and mobile use, which make instant communication within the team possible even when team members are out of their normal working area.

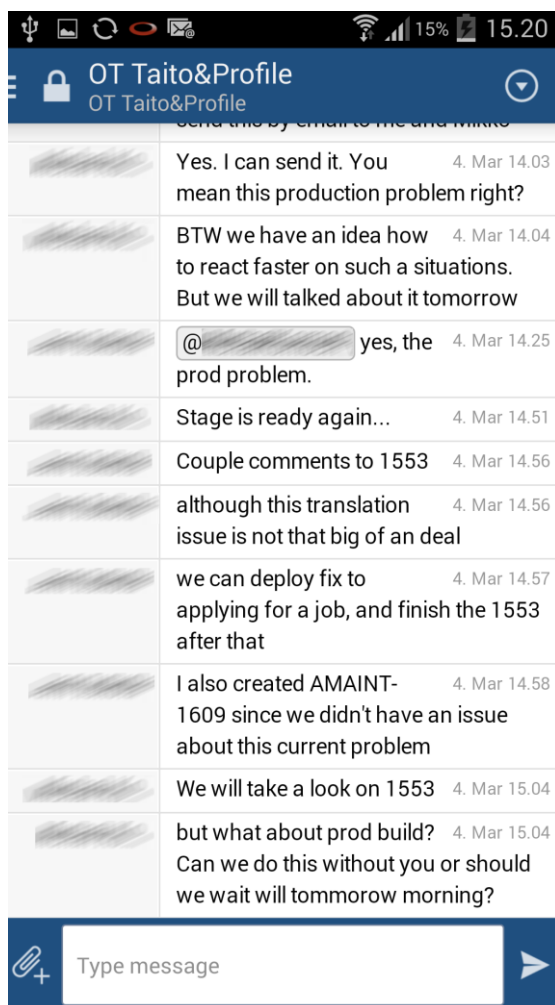


Figure 17. Instant messaging in a mobile client

6 Results

Answers to the research questions are presented below.

- 1) The team had performance issues relating to time, quality and operational issues. These issues and how they were felt having to have been affected by implementing Scrum are presented in the following chapters in more detail.
- 2) Based on the SWOT and the conceptual framework regarding virtual teams and software development methods, Scrum was seen as the best method to solve the performance issues.
- 3) Based on the survey after implementing Scrum and maintaining it for 5 iterations, it is suggested that the chosen method did help improve performance. Effects of implementing Scrum are presented in the following chapters in more detail.

Starting from the end of June, the team followed the Scrum development method. All aspects of Scrum were discussed in detail with all team members, feedback from the team was asked on how to name different steps, what the sprint length should be and how to arrange the weekly and daily meetings and other Scrum related routines. After ten weeks and 5 sprints, the same survey, as before implementing Scrum, was sent to all team members, in order to evaluate the effect of Scrum on the issues that were felt problematic. In the following the results of the second survey are discussed. Weighted average from the June survey is added as a column, as is the change between the weighted averages between the two measuring points. If a change is considered positive, the Δ -value is presented with a green background colour.

The team composition did not change in any ways during the implementation.

6.1 Time related issues

Table 7. Time related issues, September

Time related issues	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	WAVG (SEPT)	WAVG (JUNE)	Δ WAVG
It takes too long for an issue to become a working, released feature in general	4	3	1	1		1,9	2,6	0,7
It is difficult to predict with any amount of certainty when an issue is ready after it has been specified	3	6				1,7	2,9	1,2
During the implementation, unforeseen problems usually occur that will delay the finishing of the issue	1	5	2	1		2,3	3	0,7
When a deadline for a certain feature is set, it usually fails	2	7				1,8	2,2	0,4

After Scrum, it is notable that all areas of the time related issues were felt to have been improved. Biggest change was in being able to predict with certainty when an issue is ready. Similarly the team felt that after Scrum, issues took shorter time to become a working issue than before it.

6.2 Quality related issues

Table 8. Quality related issues, September

Quality related issues	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	WAVG (SEPT)	WAVG (JUNE)	Δ WAVG
After an issue is done and released to production often something that previously worked, breaks down.	2	6	1			1,9	2,2	0,3
Often even after smoke tests, internal revision and quality assurance, the issue might still not work as planned or is broken	1	7	1			2	2,2	0,2
It is difficult to measure the quality of the released issue		7	2			3,2	2,4	0,8
If I am the owner of the issue, it usually correspond well to my specifications once it is returned to me as done			1	7	1	4	3,2	0,8
I need more information about the overall business strategy and client needs to do my part properly	1	4	3		1	2,6	2,8	0,2
If I need help, I know who to ask and I get the information I need to proceed					9	5	4,7	0,3
I understand the overall development workflow well				2	7	4,8	3,8	1
I understand what is expected from me as a part of the process			1	8	9	4,9	3,7	1,2

The findings for quality related issues were similar to the previous issues. All areas were felt to have been improved. Biggest effect of Scrum in this area was that there was a significant improvement in understanding what the responsibilities were for individual team members after an issue was assigned. The team also felt that they understood the overall development workflow better, the issues being done corresponded better their original specification and it was easier to measure the quality of issues than before Scrum.

6.3 Operative issues

Table 9. Operative issues, September

Operative issues	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	WAVG	WAVG (JUNE)	Δ WAVG
Issue prioritization is erratic; it changes so often that it has a negative impact in my work	4	5				1,56	2,3	0,74
Reasoning behind changing prioritization is not clear enough	3	6				1,6	2,1	0,5
Reasoning behind the issues (why something is done) is so unclear that it has negative impact in my work	3	6				1,7	2	0,3
Once I get an issue assigned to me, I understand what is needed from me				7	2	4,2	3,2	1
Once an issue is rejected and returned to me, I understand clearly why				7	2	4,2	4	0,2
At this point of the product's lifecycle, I feel it is more important to release new features fast		3	5	1		2,9	2,4	0,5
At this point of the product's lifecycle, I feel it is more important to release fewer features, and use more time to make sure that the features work well			3	4	2	3,9	4,3	-0,4

Findings for operative issues followed the two previous area of development. The team felt that all areas improved. There were two questions about if the team felt that the product should now be developed fast or slower with better quality. There is no desired state for these questions as there are for the other questions, so it can't be said that these are either "good" or "bad" developments. There is a slight indication that after Scrum, the team feels that it would be better to do things faster. This can be seen logical, because the visibility of issues being done has increased and the team has gotten used to evaluate their own performance with a better confidence. After Scrum, the team felt that the prioritization of issues was better, reasoning behind the issues was better and the most improved area was that team members understood much better what was expected from them after issues was assigned to them.

7 Conclusions

To make virtual teams work, it is important to understand what their typical challenges are and how they differ from a conventional, co-located team. Virtual teams seem to have a more pronounced need for building and instilling trust than co-located teams. In co-located teams when all team members are physically in the same space, it is more natural to assign tasks and follow their progress and observe the team spirit in general. Co-located teams might suffer from performance issues and the atmosphere in judging their importance might be more forgiving than in a virtual team. In a co-located team, the team leader might create more personal relationships with the team members and may be more willing to overlook problems that would not be overlooked in a virtual team. Instilling trust in a co-located team happens more implicitly, as a by product of the daily work. The team leader sees very closely how the team behaves and how it performs. If problems occur, they can be addressed right away to make sure they are fixed.

In virtual teams, where all natural interactions between team members are limited, the feeling of isolation can create an atmosphere where the team members do not feel themselves as a part of a group as they do in a co-located team. In order to function properly, the team members require absolute knowledge on what their responsibilities are, what they are supposed to be doing and how things should be done. Co-located team members need this knowledge too, but it is gained through both formal and

informal actions as a part of their daily work. If the spoken and written language is different between the team members, defining the ways to work and ensuring that all understand them in the same way is even more important.

Findings of this study suggest that Scrum can be successfully used to improve at least the perceived performance of a virtual team. Scrum has capabilities that provide answers to challenges in virtual teams. According to the results, Scrum can provide visibility across the team to focus on what needs to be done by using backlogs, helps to understand the workflow by giving very explicit ways of how to communicate on different issues inside the team while work is being done and creates trust by creating a working culture where everyone knows what and how work should be done. The results of this study suggests that following Scrum helps making the work the team is doing more visible for all members. Team members demonstrate what they have achieved within the team to each other and to the product owner, who can see the results right away. In time the product owner will come to know what is the best way to make the developer understand what is wanted. Developers will have a better understanding of what the actual business issue is and how their work is a part of the bigger picture. By promoting these areas, Scrum helps in instilling trust within the team.

Implementing Scrum improved the performance of the virtual team researched in this thesis. Even though the team did not feel that their situation was bad in the start, Scrum seemed to increase the performance in all observed issues. Atmosphere within the team became better, as the Product Manager and Production Manager were tied more closely to what the team was doing. Interacting with customers became easier, as the business was now able to use velocity with a calculable certainty what would be ready in the next two to four weeks.

Implementing a virtual team requires commitment which differs from working with a co-located team. We work naturally in teams when we are in the same space, but this is not the case in virtual teams. Understanding this is the first step to make them work.

References

Alliance, A., 2014. *The Agile Manifesto*. [Online]

Available at: www.agilealliance.org/the-alliance/the-agile-manifesto

[Accessed 26 10 2014].

Anderson, D. J., 2010. *Successful Evolutionary Change for Technology Organizations*, s.l.: s.n.

Atlassian, 2014. *Atlassian products*. [Online]

Available at: <https://www.atlassian.com/software>

[Accessed 29 10 2014].

Beck, K. ym., 2001. *Manifesto for Agile Software Development*. [Online]

Available at: <http://agilemanifesto.org>

[Accessed 26 10 2014].

Biju, S. M., 2010. Agile Software Development Methods and its Advantages. In: *Technological Developments in Networking, Education and Automation*. s.l.:Springer Netherlands, pp. 603-607.

Cameron, E. & G. M., 2012. *Making sense of change management*. 3 edit. s.l.:Kogan Page.

Collins, J., 2012. *Research Skills & Methods*, Helsinki: Metropolia University of Applied Sciences.

Deming, E. W., 1986. *Out of the Crisis*. 1 edit. s.l.:The MIT Press.

Devi, V., 2013. *Traditional and Agile Methods: An Interpretation*. [Online]

Available at: <https://www.scrumalliance.org/community/articles/2013/january/traditional->

and-agile-methods-an-interpretation

[Accessed 11 10 2014].

Dooley, J., 2011. *Software Development and Professional Practice*. 1 edit. New York: Springer Science & Business Media.

Easterby-Smith, M., Thorpe, R. & Jackson, P. R., 2008. *Management Research*. 3 edit. s.l.:Sage Publications.

Everyday Kanban, 2012. *Everyday Kanban*. [Online]

Available at: <http://www.everydaykanban.com/what-is-kanban/>

[Accessed 19 10 2014].

Hungwei, T. & Heng-Yu, K., 2011. The relationship between trust, performance, satisfaction and development progressions among virtual teams. *The Quarterly Review of Distance Education*, 2(12), pp. 81-94.

Inoa, 2014. *Oikotie liikevaihto*. [Online]

Available at: <http://www.finder.fi/IT-sovelluksia,%20IT-ohjelmistoja/Oikotie//taloustiedot/379134>

[Accessed 8 11 2014].

Johnson, G. W. R. & S. E., 2009. *Fundamentals of strategy*. 2 edit. Essex: Pearson Education Limited.

Katzenbach, J. R. & S. D. K., 2002. *The Wisdom of teams*. s.l.:Collins Business.

Kotter, J. P., 2002. *The Heart of Change*. s.l.:Harvard Business School Press.

Lencioni, P., 2012. *The Advantage. Why Organizational Health Trumps Everything Else in Business*. s.l.:Jossey-Bass.

Library Technology Reports, 2008. Best Practices for Working in a Virtual Team Environment. *Library Technology Reports*, Vol. 2008, pp. 28-30.

MacDonnel, R., O'Neill, T., Kline, T. & Hambley, L., 2009. Bringing Group-Level Personality to the Electronic Realm: A Comparison of Face-to-Face and Virtual Contexts. *The Psychologist-Manager Journal*, 2009(12), pp. 1-24.

Mitch Lacey & Associates, 2014. *Sprint Review*. [Online]
Available at: <https://www.mitchlacey.com/intro-to-agile/scrum/sprint-review>
[Accessed 26 10 2014].

Narayan, R., 2008. Work Dispersion, Process-Based Learning and Offshore Software Development Performance. *MIS Quaterly*, 32(2).

Nunamaker, J. F., Bruce, R. A. & Briggs, R. O., 2009. Principles for Effective Virtual Teamwork. *Communications of the ACM i*, 2009(4), pp. 113 - 117.

Oliveira, C., 2014. *How to Calculate and Use Velocity to Help Your Team and Your Projects*. [Online]
Available at: <https://www.scrumalliance.org/community/articles/2014/february/velocity>
[Accessed 29 10 2014].

Panteli, N. & Tucker, R., 2009. Power and Trust in Global Virtual Teams. *Communications of the ACM*, 12(52), pp. 113-115.

Pichler, R., 2007. *Being an Effective Product Owner*. [Online]
Available at: <https://www.scrumalliance.org/community/articles/2007/april/being-an-effective-product-owner>
[Accessed 26 10 2014].

Pichler, R., 2010. *Agile Product Management with Scrum*. Boston: Pearson Education .

Royce, W. W., 1970. *Managing the Development of Large Software Systems*, s.l.: The Institute of Electrical and Electronis Engineers.

Schein, E. H. & B. W. G., 1967. *Personal and Organizational Change Throught Group Methods*. 1 edit. s.l.:John Wiley & Sons Ltd.

Schwaber, K. & Beedle, M., 2002. *Agile Software Development with Scrum*. s.l.:Prentice Hall.

Scrum Alliance, 2014. *Core Scrum*. [Online]
Available at: <https://www.scrumalliance.org/why-scrum/core-scrum-values-roles>
[Accessed 26 10 2014].

Scrum Alliance, 2014. *Who is Scrum Alliance*. [Online]
Available at: <https://www.scrumalliance.org/about-us>
[Accessed 26 10 2014].

Scrum Alliance, 2014. *Why Scrum*. [Online]
Available at: <https://www.scrumalliance.org/why-scrum>
[Accessed 10 19 2014].

Shore Labs, 2014. *Kanban Tool*. [Online]
Available at: <http://kanbantool.com/what-is-kanban>
[Accessed 19 10 2014].

Tuffely, D., 2012. Optimising virtual team leadership in Global Software Development. *The Institution of Engineering and Technology*, 6(3), pp. 176-184.

Y. Sugimori, K. K. F. C. & S. U., 2007. Toyota production system and Kanban system. *International Journal of Production Research*, 15(6), pp. 553-564.

Yin, R. K., 2003. *Case Study Research, Design and Methods*. 3 edit. s.l.:Sage Publications.

Appendix 1 Questionnaire on performance issues

Hello,

in order to improve the development cycle of the Grimstad (Taito&Profile)-project from an idea to a released feature we are now gathering data on what kind of issues are seen as problematic by different parties.

As a part of this, I am asking you to list (in a free format, bullet points, short descriptions etc are fine) issues regarding any aspect of the daily development work that you feel is problematic.

Please send me the answers via email by the end of the next week (Friday, may 9th)

This questionnaire is sent to all relevant stakeholders working with the Taito&Profile development. Answers are analyzed and collated, possibly followed by a more detailed survey. After this we will proceed on planning how to change the actual daily development process. Your answers are extremely important in order to improve the process. All answers are handled confidentially.

If you have any questions about this questionnaire, please contact me via email/hipchat/phone.

BR Mikko

Mikko Takkunen
Development Manager, Team Leader

Oikotie Oy
Töölönlahdenkatu 2
PL 35, 00089 SANOMA

Sanoma.com

This is an e mail from Sanoma Corporation and is intended only for the addressee.
The email may contain privileged and confidential information.
If you receive this email by mistake, please return it to Sanoma Corporation without producing, distributing or retaining copies thereof.
Thank you.

Appendix 2, June and September Surveys

Evaluating the Grimstad (Taito&Profili) software development process

1 / 4

25%

Background and context information

1. The country I work in

- Finland
- Poland

2. The function I work in

- Production management
- Production (software) development
- Business

3. I have been working with the project/service for

- less than a year
- 1-2 years
- more than 2 years

4. My daily allocation for this project / service is about

- 0-10%
- 11-50%
- 51-80%
- 81-100%

5. Time related issues

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
It takes too long for an issue to become a working, released feature in general	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It is difficult to predict with any amount of certainty when an issue is ready after it has been specified	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
During the implementation, unforeseen problems usually occur that will delay the finishing of the issue	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
When a deadline for a certain feature is set, it usually fails	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. Quality related issues

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
After an issue is done and released to production often something that previously worked, breaks down.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Often even after smoke tests, internal revision and quality assurance, the issue might still not work as planned or is broken	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It is difficult to measure the quality of the released issue	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
If I am the owner of the issue, it usually correspond well to my specifications once it is returned to me as done	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I need more information about the overall business strategy and client needs to do my part properly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
If I need help, I know who to ask and I get the information I need to proceed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I understand the overall development workflow well	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I understand what is expected from me as a part of the process	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. To me, quality mostly means

- How well the released issue corresponds the original specification
- How well the released issue follows the best technical practices in efficiency, security etc
- The money the customer is ready to pay for the feature

Other (please specify)

8. Operative issues

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
Issue prioritization is erratic; it changes so often that it has a negative impact in my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reasoning behind changing prioritization is not clear enough	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reasoning behind the issues (why something is done) is so unclear that it has negative impact in my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Once I get an issue assigned to me, I understand what is needed from me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Once an issue is rejected and returned to me, I understand clearly why	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
At this point of the product's lifecycle, I feel it is more important to release new features fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
At this point of the product's lifecycle, I feel it is more important to release fewer features, and use more time to make sure that the features work well	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>