



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JAANA SEPPÄNEN
SCRUM – FROM THEORY TO PRACTICE IN SOFTWARE
DEVELOPMENT

Master of Science Thesis

Examiner: prof. Tommi Mikkonen
Examiner and topic approved by the
Faculty Council of the Faculty of
Information Technology
on 6th April 2016

ABSTRACT

JAANA SEPPÄNEN: Scrum – from Theory to Practice in Software Development
Tampere University of Technology
Master of Science Thesis, 54 pages, 7 Appendix pages
June 2016
Master's Degree Programme in Information Technology
Major: Software Engineering
Examiner: Professor Tommi Mikkonen

Keywords: agile, Scrum, Scrum practices, questionnaire survey

Scrum is the most used agile methodology and its popularity is growing. More and more companies and teams use Scrum, but there are not many scientific studies done showing what *Scrum practices* are actually used and how those practices have been employed.

The purpose of this thesis is to provide more information on this topic. Research questions for this thesis are: 1) What Scrum practices are used on the field and how they have been employed? 2) What engineering practices are used along with them? 3) What are the positive and negative sides of using Scrum? and 4) What are the outcomes of using Scrum?

In this thesis the background and history of agile and Scrum are introduced. The basic terms and practices of Scrum are presented. To research the problem and get data from the field articles were researched systematically to collect information from earlier studies. In addition a questionnaire was formed to get more information on the real usage of Scrum. Questionnaire was sent to several different persons working as a part of a Scrum Team.

This thesis presents the results of how Scrum is used in real life compared to textbook Scrum practices, and suggests what kind of improvements can be made based on the findings. The thesis also points out many positive sides of Scrum, highlighting the actual benefits of using it. It also presents challenges and perceived difficulties with Scrum.

TIIVISTELMÄ

JAANA SEPPÄNEN: Scrum – from Theory to Practice in Software

Development

Tampereen teknillinen yliopisto

Diplomityö, 54 sivua, 7 liitesivua

Kesäkuu 2016

Tietotekniikan diplomi-insinöörin tutkinto-ohjelma

Pääaine: Ohjelmistotuotanto

Tarkastaja: professori Tommi Mikkonen

Avainsanat: agile, Scrum, ketterät menetelmät, ohjelmistotuotantomenetelmä, kyselytutkimus

Scrum on tällä hetkellä ehkä käytetyin ketterä menetelmä, ja sen suosio kasvaa koko ajan. Yhä useampi yritys ja tiimi käyttää Scrumia, mutta tästä huolimatta saatavilla ei juurikaan ole tieteellisiä tutkimuksia siitä, mitä Scrum-käytäntöjä itse asiassa käytetään ja miten ne on otettu käyttöön erilaisissa organisaatioissa.

Tämä diplomityön tarkoituksena on tuoda lisää tietoa tähän aihepiiriin. Työn tutkimuskysymykset ovat: 1) Mitä Scrum-käytäntöjä yrityksissä oikeasti hyödynnetään, ja kuinka ne on otettu käyttöön? 2) Mitä muita työmenetelmiä Scrumin ohella käytetään? 3) Mitkä ovat Scrumin käytön tunnistetut positiiviset ja negatiiviset puolet? ja 4) Mitä hyötyjä tai haittoja Scrumin käytöstä on seurannut?

Ongelman tutkimista varten suoritettiin systemaattinen kirjallisuustutkimus, jolla saatiin aineistoa aiemmin raportoiduista tuloksista. Lisäksi luotiin kysely, jolla haluttiin selvittää Scrumin käyttöä yrityksissä. Kysely lähetettiin vastattavaksi usealle henkilölle jotka työskentelevät osana Scrum-tiimiä.

Tulokset kuvaavat, miten Scrumia käytetään yritysmaailmassa. Tulosta verrataan Scrum-ohjeistuksissa annettuihin periaatteisiin. Lisäksi otetaan kantaa siihen, kuinka vastaajat voisivat parantaa työskentelytapojaan Scrum-teoriaan pohjautuen. Diplomi-työssä myös esitetään Scrumin positiivisia puolia ja mitä hyötyjä se tuo tullessaan. Lisäksi listataan ongelmakohtia, ja raportoidaan mitkä asiat koetaan vaikeaksi Scrumin mukaisesti työskenneltäessä.

PREFACE

This Master thesis is presented to the Faculty of Software Engineering at Tampere University of Technology. In this thesis I collect information with a questionnaire sent to several Scrum Team members – my aim is to research how Scrum is actually used in practice in Software Development, what engineering practices are used alongside of it, and how the usage of Scrum is perceived by Scrum Team members.

Questionnaire answers reveal that in general feelings about Scrum are positive, and find many good things accompany its use, like better team collaboration, task and progress transparency, continuous learning, and delivery of better quality code. Interestingly, most of the teams do not use all the artifacts and Scrum practices as prescribed in textbook Scrum. Teams are taking parts that fit their use or they have modified how they use them.

I combine collected data and link it to the perceptions of the respondents to find parts that are not working so well and make suggestions for improving the work practices of the respondents. I also gathered comparable data to compare findings to other studies made in the field.

Here are some of the comments from respondents to describe the opinion and attitude towards Scrum: *“it gives good overview and feeling of progress”*, *“We are currently very relaxed in the way we are doing Scrum”* and *“Today everybody participate and share opinions even on areas that they normally don't know about. It helps us despite the time used - it is a lot easier to share tasks and help each other”*. Some challenges are still faced with working with Scrum: *“We are missing a Scrum Master that are enforcing good Scrum practices”*, *“Hard to plan tasks good enough long time ahead - often we only plan for near future (depending on the project of course)”*. Also customers are not taken into the Scrum process or they are against it *“They do not accept that we are planning in 2week steps”*, *“customer doesn't know”* and *“Mainly quality people care as they can look at statistic”*.

First I would like to thank Professor Tommi Mikkonen for his guidance and help during writing of this thesis. I would also like to thank all those people who helped me finalize my thesis by answering my questionnaire. Lastly I would like to express my deepest gratitude to my mother, my sister, and foremost Juan Pedro: it would have not been possible to do this without you; thank-you for all your time and help.

Tampere, 18.5.2016

Jaana Seppänen

CONTENTS

1.	INTRODUCTION	1
2.	AGILE SOFTWARE DEVELOPMENT AND SCRUM.....	2
	2.1 History of agile.....	2
	2.2 Scrum	3
	2.2.1 Scrum at glance.....	3
	2.2.2 Scrum Team	5
	2.2.3 Scrum artifacts	5
	2.2.4 Scrum events	6
	2.2.5 Related Scrum terms	7
	2.3 Scrum versus traditional development models.....	8
3.	BACKGROUND MATERIAL AND RESEARCH METHOD	9
	3.1 Research method	9
	3.2 Research on background material	10
	3.3 Formation of the Questionnaire.....	12
4.	SCRUM IN PRACTICE	14
	4.1 Practices used in the companies according to the articles.....	14
	4.2 Positive and negative sides of Scrum according to the articles.....	15
	4.3 Modified Scrum according to the articles	17
	4.4 Questionnaire findings	18
	4.4.1 General overview of the respondents.....	18
	4.4.2 Scrum Team	20
	4.4.3 Scrum artifacts	22
	4.4.4 Scrum events	27
	4.4.5 Engineering practices used.....	32
	4.4.6 Questionnaire findings on tools usage	32
	4.4.7 Testing.....	34
	4.4.8 Positive and negative aspects of Scrum	35
	4.4.9 Actual outcomes seen by the teams	39
	4.5 How do questionnaire results compare to previous studies	40
	4.6 Tools comparison with questionnaire and articles	42
	4.7 Perceptions of using Scrum by articles and questionnaire.....	42
5.	EVALUATION.....	45
	5.1 Research questions revisited	45
	5.2 Uncertainty factors	46
	5.3 Limitations of the research.....	47
	5.4 Future work	47
6.	CONCLUSIONS.....	49

APPENDIX A: TEMPLATE FOR DATA COLLECTION FROM ARTICLES

APPENDIX B: QUESTIONNAIRE

LIST OF FIGURES

<i>Figure 1 History of agile [18].....</i>	<i>3</i>
<i>Figure 2 Scrum at glance [32].....</i>	<i>4</i>
<i>Figure 3 Empirical process model for Scrum [31].....</i>	<i>4</i>
<i>Figure 4 Overview of respondents' role in Scrum Team.....</i>	<i>18</i>
<i>Figure 5 Respondents' Scrum experience.....</i>	<i>19</i>
<i>Figure 6 Size of Scrum Teams.....</i>	<i>20</i>
<i>Figure 7 Team location local vs distributed.....</i>	<i>21</i>
<i>Figure 8 Scrum Master helping to achieve Sprint goal.....</i>	<i>22</i>
<i>Figure 9 Product Backlog related findings.....</i>	<i>23</i>
<i>Figure 10 Usage of Burndown Chart.....</i>	<i>24</i>
<i>Figure 11 Tasks and team interruptions.....</i>	<i>25</i>
<i>Figure 12 Sprint duration.....</i>	<i>27</i>
<i>Figure 13 Duration of Daily Scrum.....</i>	<i>28</i>
<i>Figure 14 Sprint Backlog contents.....</i>	<i>30</i>
<i>Figure 15 Engineering practices used.....</i>	<i>32</i>
<i>Figure 16 Communication tools.....</i>	<i>33</i>
<i>Figure 17 Testing tools.....</i>	<i>33</i>
<i>Figure 18 Agile project management tools.....</i>	<i>34</i>
<i>Figure 19 Different ways of testing used in respondent's companies.....</i>	<i>34</i>
<i>Figure 20 Outcomes of Scrum.....</i>	<i>40</i>
<i>Figure 21 Comparison of practices: articles, questionnaire and Vuorinen thesis.....</i>	<i>41</i>

LIST OF TABLES

<i>Table 1 Scrum vs traditional development models</i>	<i>8</i>
<i>Table 2 Background Articles.....</i>	<i>11</i>
<i>Table 3 Respondent general overview</i>	<i>19</i>
<i>Table 4 Perceptions of Scrum</i>	<i>36</i>

1. INTRODUCTION

Scrum is the most used agile process: according to a 2013 Forrester study, 90% of agile practitioners use Scrum [10]. 2015 VersionOne survey results mention 58% of agile users are using Scrum [45]. Despite the 30% variance in these sources, one can nevertheless state that Scrum is very popular and more and more companies are starting to use it [45]. Scrum is advertised to bring multiple benefits such as: better teamwork, focus on customer needs, flexibility and adaptability, better quality code, and more satisfied developers [31]. Several challenges such as architectural design, lack of management support, and lack of knowledge of Scrum are also reported by different studies [3], [9], [26], [28]. Multiple studies also show that modifications are made to the way Scrum is used, which brings us to the inspiration for this thesis [3], [8], [9], [27], [42].

The aim of this thesis is to provide more knowledge on topic of what Scrum practices are used, and if they are being used as textbook Scrum defines them. Based on the responses, recommendations are made to improve Scrum implementation. This thesis poses the following research questions:

- ***What Scrum practices are used in the field and how they are employed?*** There are very few studies which establish what Scrum practices are actually in use, and whether or not they are used as prescribed in textbook Scrum. This research question is trying to provide more data in this area.
- ***What engineering practices are used along with them?*** Scrum requires a working environment where each Sprint needs to deliver a working piece of software. This sets some requirements for example on testing, integration, and delivery of software. This thesis tries to find out what engineering practices used with Scrum enable continuous software delivery with good quality.
- ***What are the positive and negative sides of using Scrum?*** This question seeks more insight as to why some practices are better adopted than others. Asking for users' views about using certain Scrum practices gives information as to why they work or don't work.
- ***What are the outcomes of using Scrum?*** This question tries to clarify what the benefits of using Scrum are.

In Chapter 2 the history of agile and Scrum and the basic Scrum methods are represented. In Chapter 3 thesis research methods are explained, and a description of article research and questionnaire formation is provided. In Chapter 4 the findings from the background articles and questionnaire are presented. Chapter 5 evaluates the study questions, limitations of the research, future research possibilities and uncertainty factors. In Chapter 6 the thesis findings are concluded.

2. AGILE SOFTWARE DEVELOPMENT AND SCRUM

2.1 History of agile

In the history of software development and evolution, the first software conference was held in 1968 by the NATO Science Committee. A more process-like approach to software development was needed to replace ‘ad hoc’-like software manufacturing. The principles agreed on this conference later led to the introduction of a waterfall model by Royce in 1970s. The waterfall model is still used today by many companies. [25]

In the 1970s product development projects followed a phase-to-phase process: first a concept phase, then a feasibility study followed by design, development, pilot production, and final production. [17]

The new game was on. Companies’ sales were coming from the ‘new products’ on the market, and companies needed to be more flexible and get products out faster. They needed to produce more high-quality products with lower cost. In January 1986, the *Harvard Business Review* released an article by Hirotaka Takeuchi and Ikujiro Nonaka which introduced a new holistic process to emphasize speed and flexibility in product development. This process followed a rugby approach wherein the team proceeds as a one unit and the core of the process has six characteristics: built-in instability, self-organizing project teams, overlapping development phases, multilearning, subtle control, and organizational transfer of learning. This process was called Scrum. [17]

“Agile” is an umbrella term for set of different practices that specify an iterative approach to software development. Scrum is one type of an agile method, though often it is used as synonym for agile. Other agile methods are Extreme Programming, Feature Driven Programming and Crystal, just to mention few. In 2001, a group of people who presented these agile development processes gathered together to discuss common principles; the outcome was the Agile Manifesto, a set of principles to govern agile software development. [16]

Here are the principles that all agile methods should follow, Agile Manifesto [5]:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.
- That is, while there is value in the items on the right, we value the items on the left more.

Figure 1 presents the history of agile on a timescale from NASA Iterative Incremental Delivery 1960 all the way up to the Agile Manifesto in 2001. Since the manifesto, the usage of agile practices have been increasing year after year [45].

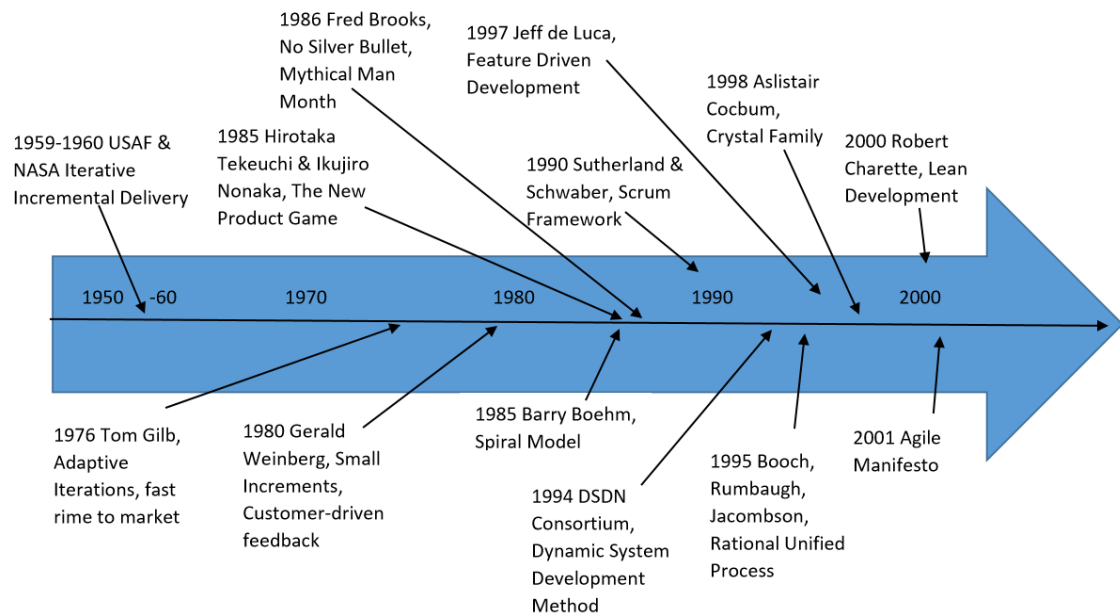


Figure 1 History of agile [18]

2.2 Scrum

“Scrum represents a new, more accurate way of doing software development that is based on the assumption that software is a new product every time that is written or composed” [31]. Scrum’s idea originally comes from a Japanese development process, where Scrum term refers to rugby star formation’s idea to progress as a unit over the whole field. Scrum process and rugby strategy are both self-organizing and adaptive. [31]

2.2.1 Scrum at glance

Figure 2 shows the core of Scrum and all of its central terms [36]:

- Scrum artifacts: Product Backlog, Sprint Backlog, Increment and Burndown Chart.
- Scrum Team: Development Team, Scrum Master and Product Owner.
- Scrum Events: Sprint, Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective.

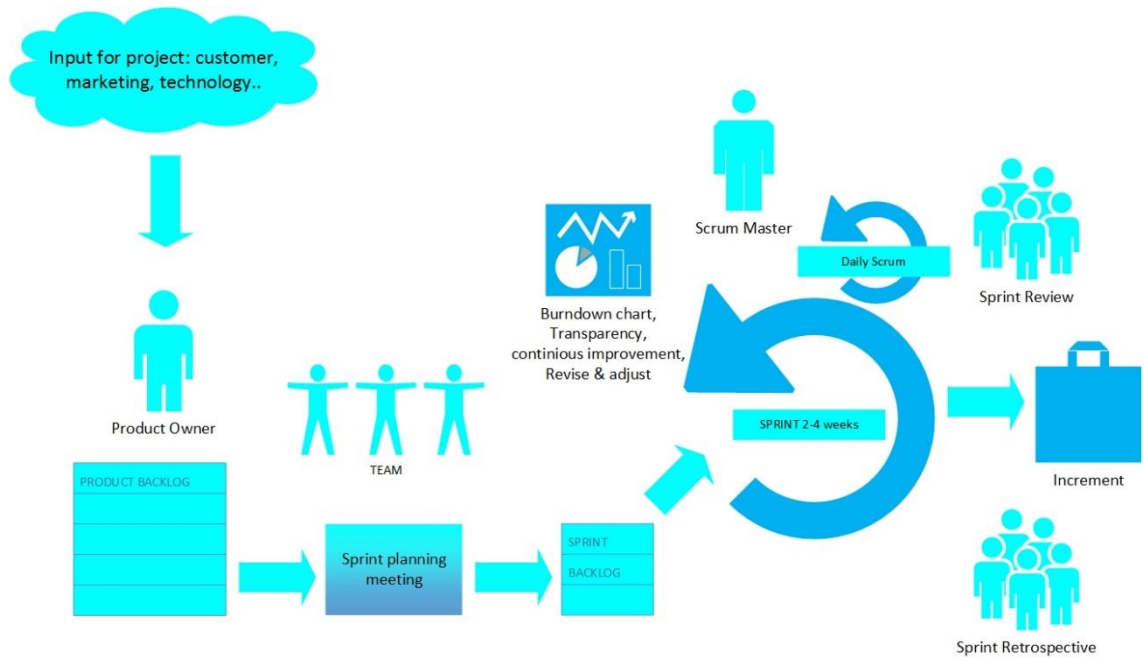


Figure 2 Scrum at glance [32]

The three pillars of Scrum are: 1) transparency, common language and definition; 2) inspection; inspect Scrum artifacts progressing towards the Sprint Goal; and 3) adaptation; if in inspection it is noticed one or more aspects are not within limits, the process needs to be adapted to fix the course towards the accepted area.[36]

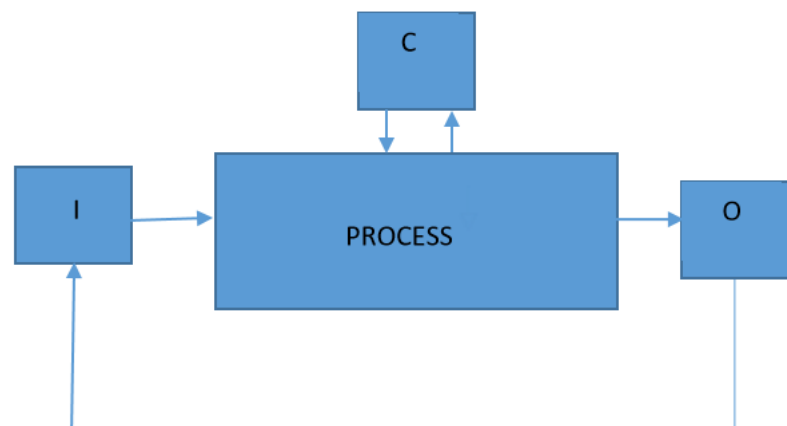


Figure 3 Empirical process model for Scrum [31]

Scrum is an empirical process model where: an input (I) team has the requirements, output (O) is the Increment, and the control to the process (C) is the Daily Scrum and Review meetings whereby the team empirically determines how to build the product Increment during the Sprint (see *Figure 3*). The project is inspected, assessed and adjusted frequently, so it is more adaptable for change and unexpected situations. [31] “*Empiricism*

asserts that knowledge comes from experience and making decisions based on what is known” [36].

2.2.2 Scrum Team

Scrum Team. Team size should be seven people, plus or minus two, including the Product Owner and Scrum Master [13], [31]. Too big of a team is hard to handle in Daily Scrum meetings, and too small team size limits interaction. Scrum Teams should have all the skills and experience to meet the Sprint goal, so the team should be cross functional to include coding, testing, quality control, design and analysis. On the Scrum Team there are no titles or expectations other than everyone should do their best to reach the Sprint goal. There might be some persons that are only working part-time for the team; as system administrators, they need to know the working hours targeted for the project and keep them in mind when making commitments to the project. The team has full authority to decide how it does the work, but members are also responsible for using any existing standards or technologies to be understood by other teams and in line with other organizational products. Scrum Teams also decide which items from Product Backlog are taken into a Sprint, and what are the tasks they will commit to do within the Sprint. [31]

Scrum Master. The Scrum Master (SM) is a driving force to set up the Scrum practices, and the person to ensure they are being followed. The role of a Scrum Master is a management role: he is responsible for forming the Scrum Team along with the higher management. SM is the person who is in contact with the Product Owner, who forms the Backlog together with the Development Team. Scrum Master is responsible to listen the team, remove impediments and follow the actual progress against the expected progress. He also conducts the Daily Scrums with the Development Team and ensures things proceed and decisions are made. Schwaber notes in his book the following with regard to the importance of decision making: *“It is better to proceed with some decision than no decision at all”*. [31]

Product Owner. The Product Owner (PO) makes the decisions to prioritize the Product Backlog; everyone else must respect that he is the person in charge of the Product Backlog. He also makes the estimates to complete a task; such estimates change along the way, as the work gets more detailed. Estimates are done by asking the task duration from the Development Teams, so estimates are the best guess at the moment and can vary. [31]

2.2.3 Scrum artifacts

Product Backlog. The Product Backlog is the list of features, functions, enhancements, technologies and bug fixes that present work to be done in the product; they are the requirements. This list is dynamic, changing according to customer requirements and business needs. Input to the backlog can come from many sources, like marketing, technology team and sales, just to mention a few. The Product Backlog is ordered in priority: top

items are more detailed and urgent, lower ones have less detail and are less urgent items. The Product Owner is the person to update the Product Backlog. [31]

Sprint Backlog. The Sprint Backlog is the list of tasks chosen to be finalized during a Sprint. The Sprint Planning meeting agrees on the Sprint goal and tasks to be formed into an Increment at the end of the Sprint. [31]

Increment. The Increment is what the team will deliver after the Sprint, a Product Increment. With possibly unstable technology and complex requirements the team needs to find a way to produce an Increment within one Sprint. Daily builds are a good way for the team members to see if they have made working code and to see their own progress in real life. [31]

Burndown Chart. To mark the progress during a Sprint, a Burndown Chart is used to track completed tasks and stories. After the task breakdown in the Sprint Planning meeting, the ideal burndown line is created. During the Sprint an actual graph is updated based on the tasks and stories done. [7]

2.2.4 Scrum events

Sprint is a time box to do one iteration; the recommended length is 30 calendar days, but the time of the Sprint can vary according to a team and project from 2 to 4 weeks. Sprint starts with a Sprint Planning meeting to agree on the Sprint Goal and to choose items from the Product Backlog and turn them into Sprint Backlog and further into tasks. [15]

Daily Scrum meetings are arranged by the Scrum Master, always at the same place and same time lasting 15 minutes. Every team member should join the meeting either via phone or face-to-face. In this meeting the team members answer 3 questions:

- What have been done since the last Daily Scrum?
- What will be done between now and the next Daily Scrum?
- What are the impediments?

Daily Scrum should be the only status meeting team members need to attend, and anyone who wants to know the project status can join as a listener. Daily Scrum provides current status of the project each 24 hours, increases team members' level of knowledge of the whole project, removes and identifies impediments, and emphasizes fast decision making. It is very important to keep the meeting short and informative; one must not start solving problems there, but merely inform others and let the Scrum Master know of possible impediments. The Scrum Master is responsible for arranging a place and running the meeting, as well as for making needed decisions about issues. If there are issues that need further follow-up, subsequent meetings can be arranged afterward. Team members who feel they need to discuss some issue will note in the meeting *"I would like to discuss*

about this after the Daily Scrum meeting” so that involved people can further discuss, while others can return to their work after the Daily Scrum. [31]

Sprint Planning meeting. In the Sprint Planning meeting, the Team, Product Owner, Scrum Master, and customers / users define the Sprint Goal, what functionality the Increment should have after the Sprint. First all meet up to decide on the Sprint Goal, and afterwards the team meets to decide how the Sprint Goal will be accomplished. The team is committed to accomplishing the Sprint Goal, but if it happens that the work is more difficult than expected, the team can partially implement the functionality and will review accomplished functionality with the customer in the Review meeting. All incomplete tasks and functionality will go back to the Product Backlog. [31]

Sprint Review meeting. A Review is held to conclude the Sprint and present the Increment to the customer and management. During the meeting the team shows what they have done, but no PowerPoint slides are permitted. It sticks to a walkthrough of the completed work and discusses the functionality. Also review if things correspond to the Sprint Goal decided in the Planning meeting. A Review meeting is held every 30 days (for a 30-day Sprint). The Increment is the focal point establishing further ideas for the next Sprint and what functionality can be added in the next Sprint. [31]

Sprint Retrospective meeting. Retrospective is used to learn from the last Sprint. Feedback is collected to see what was good and what impediments there were, that is to say, what the team should continue doing and what they should not. This meeting is held at the end of the Sprint by the Scrum Master and the outcome of this meeting should be carried forward as improvements for the next Sprint. Retrospective is an important part of inspection and adaptation of Scrum. [36]

2.2.5 Related Scrum terms

Velocity. Team velocity tells what the team is able to deliver during the Sprint, and mature teams should have somewhat constant velocity, enabling the ability to estimate which new features can be delivered and what backlog items can be included in the iteration in a Sprint. [40]

Story point. Story points are used to tell how demanding a certain task is. Most commonly the team classifies each task with Fibonacci numbers 1, 2, 3, 5, 13, 21, 34 to correspond the difficulty and time expected to be consumed for a task, but other sequences can be used too. This way of estimating tasks is team-specific since it only tells how one specific team has estimated the tasks according to their capabilities and understanding. [2]

User story. The user story is a short descriptive story of a certain functionality and action. It is described as follow: As a <type of user>, I want <some goal> so that <some reason> [39]. For example “*As a user I would like to be able to check my account balance in ATM*

so I know how much money I can withdraw.” Requirements are usually written as user stories.

2.3 Scrum versus traditional development models

Table 1 presents the main differences of Scrum as compared to traditional models, such as Waterfall; it addresses planning, collaboration, requirements, project controlling, documentation and retrospective aspects. All the items in the Scrum column are perceived as positive sides of Scrum bringing improvements to the working process. The difference in documentation can also be considered as a challenge, as most knowledge transfer is done via verbal meetings and not as written documentation.

Table 1 Scrum vs traditional development models

Aspect	Scrum	Traditional methodologies
Planning	Development process managed from iteration to iteration. Planning is done on different strategy levels. Assignment of tasks discussed in meetings	Development project managed in advance. Planning done using a work breakdown structure and milestones. Project manager assigns tasks in advance.
Collaboration	Flat hierarchy with self-organizing teams. Close cooperation with customer and development team throughout the project	Project manager leads the team. Collaboration with customer usually in beginning of project.
Requirements	Continuously discussed with customer	Fixed contract like document
Project controlling	Customers can evaluate a working piece of SW at the end of each sprint. Burndown chart shows daily updated summary of remaining tasks	Team members typically returns percentage of completion for milestone status
Documentation	No instruction to write down knowledge in a documentation. Instead the knowledge transfer is done via meetings	Documentation is considered to be an integral part of the development process
Retrospectives	After each sprint	Lessons learned usually at the end of project

3. BACKGROUND MATERIAL AND RESEARCH METHOD

3.1 Research method

Purpose of the thesis. The scope of this thesis is to study usage of Scrum in Software Development. The main focus will be on Scrum practices, to answer how Scrum has been brought into use, what Scrum practices are employed in the field, and what engineering practices are used with Scrum. In addition the research focuses on the positive and negative perceptions related to Scrum, as well as the outcomes of using Scrum. Often you hear people say: “*We are using Scrum, but..*” followed by a description of what adaptations and changes have been done compared to the textbook Scrum [8], [9], [11], [42].

Thesis goals. This research aims to show the practices used in the field, revealing whether Scrum is used as prescribed in the textbook or is used with some changes. The thesis also aims to explain the reasons why Scrum is used and what the outcomes of its use are. Research findings herein are compared to those in other research articles to establish correlations and divergences. In the questionnaire research, the positive and negative perceptions provide insight into why a Scrum practice does or does not work. Questionnaire feedback forms the basis of an inquiry into possible improvement proposals for Scrum practices. This thesis does not provide team-specific improvement proposals, but evaluates practices based on answers and given improvement proposals relying on textbook Scrum in cases where the real life usage differs from the textbook.

Chosen research method. The research method for this thesis is comprised of scientific article research and questionnaire field data collection. The questionnaire was chosen to make it easier for respondents to answer on their own time. In addition respondents were located in several different countries, thus it would have been difficult to arrange face to face meetings. Skype interviews were also considered, but were rejected in favor of the questionnaire, which was more efficient at getting more responses and large amounts of information within a given time frame. The questionnaire benefits are mentioned in the *research introduction guidelines* for questionnaire, and based on it the questionnaire was considered more suitable for analyzing the results [1]. With a questionnaire it is possible to get quantitative data for comparison; also, comparison to earlier studies is easier with quantitative data [1]. Interviews usually give more qualitative data, and therefore mapping answers later can be more difficult. Of course, questionnaires can also give some qualitative data from unstructured questions; such questions were included in the perceptions part of the questionnaire to ask about positive and negative aspects of Scrum.

Focus group for the questionnaire. The respondents for the questionnaires were chosen through industry contacts; respondents were contacted personally and asked to respond to the questionnaire. The respondents are from different positions, companies, and countries. The selection criteria required a person directly working with Scrum as a developer, Scrum Master or Product Owner. Since companies were not directly contacted to obtain permission for name use, company names are not mentioned in the thesis for privacy reasons.

3.2 Research on background material

Usually scientific studies regarding Scrum compare employment of Scrum in several companies or teams. It has been quite difficult to find studies that focus on actual Scrum practices and outcomes of use in companies.

For the article research of this thesis results are presented from several different studies collected from scientific databases. Not all the studies directly focus on corporate Scrum practices, but those studies at least mentioning which parts of Scrum are being used or partly mentioning Scrum practices, are used as background material. Also if these articles mention positive and negative sides of Scrum they are taken into background material list. A total of 14 different papers and one online survey report are used as background material. To collect the material, the following systematic approach was applied:

First phase – data collection from database. The first phase collected material from IEEE Xplore database with *Scrum (SOFTWARE Development)* keyword, on 16th February 2016. It gave 397 results.

Second phase – narrowing the search. The second phase narrowed down the articles, first by limiting the search to the years 2010-2016 to get results from more recent studies; this gave 292 results. Limiting the search to conference publications gave 272 results.

Third phase – elimination based on title. Further selection based on article name (and abstract with borderline cases) gave 34 results.

Fourth phase – critical evaluation of the articles based on quality criteria. In the fourth phase, articles were more thoroughly scrutinized. Articles talking about how Scrum should be used, as well as others which were not based on scientific research, were excluded. Two needed to be removed from the list since the research was in languages the thesis author did not understand (Chinese and Spanish). One research article did not have any references marked in the text, so it could not be considered a reliable source. From the articles remaining, the following process of elimination chose the final articles used in this thesis. Articles were eliminated if they did not fulfill the following quality criteria [42]:

- Is it research (not a personal opinion)?

- Is there a clear goal for the research?
- Is there a clear description of the research and clear references?
- Is there some knowledge expressed by the research gained from Scrum usage?

Result 13 articles. Information gathering from the articles used the template from Appendix 1. All articles are presented in *Table 2*.

Table 2 Background Articles

ID	Ref	Article Name	Authors	Publish Year	Type of Article
A0	[28]	Effectiveness of Scrum for Offshore Sw Development in Sri Lanka	R.K. Chandana Ranasinghe, Indika Perera	2015	Conference paper
A1	[3]	Emergence of Agile Methods: Perceptions from Sw Practitioners in Malaysia	Ani Liza Asnawi, Andrew M. Gravell and Gary B. Wills	2012	Conference paper
A2	[8]	Do Daily Scrums Have to Take Place Each Day? A Case Study of Customized Scrum Principles at an E-Commerce Company	Daniel Pauly, Bjoern Michalik, Dirk Basten	2015	Conference paper
A3	[11]	Doing Scrum Rather Than Being Agile: A Case Study on Actual Nearshoring Practice	Franz Zieris, Stephan Salinger	2013	Conference paper
A4	[37]	Investigating the Long-Term Acceptance of Agile Methodologies: An Empirical Study of Developer Perceptions in Scrum Projects	Sven Overhage, Sebastian Schlauderer	2012	Conference paper
A5	[9]	The Maturation of Agile Sw Development Principles and Practice: Observations on Successive Industrial Studies in 2010 and 2012	David Bustard, George Wilkie, Des Greer	2012	Conference paper
A6	[6]	Project management using the Scrum agile method: A case study within a small enterprise	Breno Lisi Romano, Alan Delgado da Silva	2015	Conference paper
A7	[23]	Scrum + Engineering Practices: Experiences of Three Microsoft Teams	Laurie Williams, Gabe Brown, Adam Meltzer, Nachiappan Nagappan	2011	Conference paper
A8	[42]	Scrum menetelmän käyttö Pirkanmaalaisissa yrityksissä	Jyri Vuorinen	2010	Master Thesis

A9	[12]	Survey of Agile Tool Usage and Needs	Gayane Azizyan, Miganoush Katrin Magarian, Mira Kajko-Mattson	2011	Conference paper
A10	[26]	Scrum Practice Mitigation of Global Sw Development Coordination Challenges: A Distinctive Advantage?	Paul L. Bannerman, Emam Hossain, Ross Jeffery	2012	Conference paper
A11	[27]	Scrum and Embedded Sw Development for the Automotive Industry	Ricardo Y. Takahira, Lilian R. Laraia, Frederico A. Dias, Abraham S. Yu, Paulo T. S. Nascimento, Alceu S. Carmargo Jr	2014	Conference paper
A12	[11]	Using Agile practices to solve Global Sw Development problems – A Case Study	Sarah Beecham, John Noll, Ita Richardson	2014	Conference paper
A13	[24]	Work Motivational Challenges Regarding the Interface Between Agile Teams and a Non-Agile Surrounding Organization: A case stud	Lucas Gren, Richard Torkar, Robert Feldt	2014	Conference paper
A14	[45]	10 th annual State of Agile report	VersionOne	2015	Online survey

One more survey was added to the pool of evidence by recommendation of an Agile coach; it provided percentage comparison data from an online survey made by VersionOne, *10th annual state of agile report*. All articles used are presented in *Table 2*. Additionally, another thesis was used as background material since it employed a similar topic and research method in its study of Scrum usage in Pirkanmaa in Finland.

3.3 Formation of the Questionnaire

The questionnaire used *Nokia Scrumbutt*-test as an inspiration, but since there are no longer any results available online, no direct comparison could be made [29]. Another test *How agile are you?* was also used for some ideas for the questionnaire [43]. Questions were not copied from above sources, but inspired and set guidelines for the set of questions emphasizing how well Scrum practices are being adopted and what engineering practices are being used in tandem with Scrum.

As a trial run, the questionnaire was sent to five persons for feedback. This trial was useful, since the original questionnaire was considered too long and time-consuming. It could have been difficult to get enough answers for the research with the original version. In the trial group 1 out of 5 did not answer at all, and 2 replied only after the specified deadline of returning the questionnaire.

By recommendation of the Agile coach, some questions related to the actual outcomes of using Scrum were added, and the format of some questions was changed from unstructured to multiple choice. [45] After the test trial and review, the initial version of the questionnaire was changed to contain fewer questions and be available online in SurveyMonkey for easier respondent access [35]. Multiple-choice question format, made answering it faster and easier. Practices and outcomes questions were mostly in multi-option format, to give easily comparable data. Perception questions were unstructured open questions so respondents could tell what they thought about a specific subject. There were 19 questions in the final questionnaire version. With the second version of the questionnaire plus the answers from trial version, a total amount of 14 responses was received.

Questionnaire contents. The questionnaire had 19 questions; below is the outline of principal parts of the questionnaire. The full questionnaire is in Appendix 2.

- **General overview.** This part of the questionnaire sought each respondent's role, experience and location and team size. (Question 1)
- **Perceptions of using Scrum.** This section established what the perceptions of the respondents were towards Scrum, positive and negative, challenges and lessons learned. (Question 2)
- **Usage of Scrum practices.** Next there were questions finding out what practices were used and also some perceptions and specifics related to them. (Questions 3-15)
- **Usage of engineering practices.** This question asked what are the engineering practices used alongside Scrum practices. (Question 16)
- **Usage of tools.** This question was to clarify what kind of tools are in use. (Question 17)
- **Outcomes seen by using Scrum.** Here respondents were asked to point out all the outcomes of using Scrum. (Question 18)
- **Other comments related to Scrum.** Open comment section for any other observations related to Scrum. (Question 19)

4. SCRUM IN PRACTICE

4.1 Practices used in the companies according to the articles

Scrum artifacts. It can be concluded from the findings that not all the Scrum practices are used by all teams. Such practices not common to all were for example the Burndown Chart, mentioned only in 3 articles [A1], [A6], [A0]; in the Vuorinen thesis it was stated that 73% of the interviewed teams used it [A8]. Prioritized Product Backlog and Sprint Backlog were mentioned in 6 articles [A2], [A5], [A10], [A11], [A4], [A8]; Vuorinen stated all teams used one backlog, more specifically 55% of teams used one prioritized backlog and the remaining 45% used backlog prioritized based on ROI (return on investment) [A8].

Scrum Team. All the teams had a Scrum Team in the articles either local or distributed, a dedicated Product Owner was mentioned in 8 articles and a Scrum Master in 7 articles [A6], [A2], [A3], [A7], [A13], [A11], [A0], [A4]. The Scrum Master role was mentioned to be important and in contrary in some case the role lead into problems by SM being a bottle neck [A6], [A3]. Also the Product Owner was mentioned in the articles as having problems for example with having too many things to do simultaneously, and not having time to focus [A3].

Scrum events. Regarding Scrum events planning, Review and Retrospective meetings were mentioned in 6 papers [A6], [A2], [A3], [A7], [A11], [A0]; it was also stated in one article that the teams were using them but not for textbook Scrum purposes [A3]. The most used practice was fixed length Sprint, found in 10 articles and Vuorinen states in his thesis 91% of teams used fixed length Sprint [A1], [A6], [A2], [A3], [A7], [A13], [A11], [A0], [A4], [A8]. Daily Scrum was mentioned in 9 articles [A1], [A2], [A3], [A7], [A13], [A11], [A0], [A4].

In the survey from VersionOne, the top 5 employed practices were Daily Scrum 83%, prioritized backlog 82%, short Sprints 79%, Retrospective 74%, and the Planning meeting 69% [45.]. So all except one of most employed practices were Scrum events according to VersionOne.

Engineering practices. Few articles mentioned the engineering practices used alongside Scrum. One of the background articles researched practices affecting positively the off-shore Scrum Team [A0]. The practices they used were Daily Scrum, Scrum Retrospective Review and planning, PO and SM managing the Product Backlog and motivating the team, iterations, burn down charts, product demos and continuous integration. All these activities were found to have positive effects and contribute to more successful projects. The issues seen with Scrum were the lack of specific engineering practices such as coding

practices, automation in integration and testing and continuous integration. [A0] Another paper researched specific engineering practices with 3 Microsoft teams and found that Planning Poker, continuous integration, unit test driven development, quality gates, source control, code coverage, peer review, static analysis tools and xml documentation were helping the Scrum process and giving good results, for example better code quality [A7].

Based on the articles, the most used methods alongside Scrum were Planning Poker, Kanban, continuous integration, user stories and test driven development [A1], [A6], [A2], [A7], [A0]. Similar results were seen in the survey from version one; unit testing 63%, continuous integration 50%, Kanban 39% and TDD 33% [A14].

Tools. The 2011 article researching on tool usage in companies interviewed 121 companies in 35 countries; it found that physical wall and paper were the most common tools with 26%, spreadsheet 23%. To mention agile project management tools usage for example JIRA usage was 2%. [A9], [22] Concerning tools in general, respondents were not happy with the integration in other systems, lack of custom reports, and the absence of virtual task boards; they expressed a need to move from a high level status to more detailed one. Positive findings of the tools usage were ease of use and customizability. [A9]

4.2 Positive and negative sides of Scrum according to the articles

Product Backlog. The Product Backlog was considered very positively due to its focus on customer needs and giving transparency to the project, its flexibility to adjust to new requirements and evolve during the project, and its ability to add new features along the project [A5], [A10], [A11], [A4], [A8]. Also some studies found estimation improved [A6], [A7]. Negative sides of the backlog was included lack of ease in handling nonfunctional requirements [A5]. Also long term planning could be hard, as was working with fixed priced contracts [A5], [A4], [A8].

Sprint Backlog. The Sprint Backlog gave focus to the team and commitment to the Sprint goal [A2]. The downside regarding Sprint Backlog was a tendency to plan too many items for a Sprint, which could lead to demotivating the team when it realized the Sprint goal could not be achieved [A2].

Increment. The Increment was considered the artifact that gives the best status of the project, being a deliverable product at the end of each Sprint [A5]. It gives more confidence about project success [A5], [A10]. Being one of the best indicators of progress it has some downsides too; an architectural design was found to be difficult in some articles [A5], [A4]. In one research article some of the interviewed companies were having one Sprint just for architectural design [A5]. If the definition of “done” was not clear to all, it might lead to the dilemma that at the end of the Sprint there were non-working pieces of

software or the increment did not work at all [A8]. Maintenance was found to be an issue too [A4]. Lack of documentation is also linked to the Increment [A1]. Lack of coding standards, automated integration and testing and continuous integration can also cause quality issues [A0].

Sprint. Sprint was considered to give the team a chance to inspect and adjust in short intervals and better its performance [A5], [A10]. It starts all the project phases already in the beginning of first Sprint, establishing testing, integration and delivery, giving a good overview of issues early on, and leading to better quality with less errors [A1], [A5], [A7], [A8]. Problems were seen when there were lots of external distractions of the team during the Sprint [A6].

Daily Scrum. The Daily Scrum was seen as a big part of the knowledge sharing [A6], [A13], [A10], [A4] aspect for Scrum; also studies mentioned it helped achieve better communication, focus, team spirit and new ideas [A6], [A2], [A3], [A4]. Issues found with this event were that it sometimes got too detailed on discussion and sometimes took longer than expected [A3], [A8].

Planning meeting. According to the articles the Planning meeting helped to focus on what the customer wants [A4], [A8]. It was a forum for communication with the customer [A4]. It gave transparency to the project according to the studies [A2]. Planning Poker helped to get more accurate estimations [A6], [A7]. Some of the less-liked aspects were the cases when customer did not really understand Scrum and when it was felt meetings took too much time from actual coding [A2], [A8].

Review meeting. According to findings in the articles, the Review meeting gave instant feedback to developers and they could see their work and results and feel proud; it was self-rewarding and motivating to see work results and get feedback after a Sprint [A1], [A5], [A13], [A10], [A4], [A8]. Negative sides were that there were too many meetings, especially if there were 2-week Sprints, amount of meetings is felt to be too much [A8]. Some of the teams in the studies did not have Review meetings at all [A3].

Retrospective. The Retrospective meeting was considered positive when it clarified problems from the last Sprint and adjusted work processes [A2], [A3]; it was the place to learn from mistakes [A5]. This meeting in particular was mentioned to be problematic when in some teams it did not serve the purpose of taking things on the agenda for the next Sprint to solve, but rather was just kept for the sake of having it [A3]. Another criticism was when too technical problems, not really people and process issues, were taken up there [A3].

Scrum Master. According to some articles the Scrum Master was the guarantor that the Scrum practices would be followed and that the team worked as one [A10]. The role was very important and the role in the team had a clear purpose [A3], [A13]. The Scrum Master was also the contact point for the Product Owner [A3]. Issues were seen when the

Scrum Master became a bottle neck and when Scrum Master did not have Scrum experience or knowledge [A3].

Product Owner. Findings indicate the Product Owner (PO) was the link to the customer and gave instant feedback. He also was the person to work closely with Scrum Master [A2], [A13]. The PO and SM have each clear roles and responsibilities [A2], [A0], [A3], [A13]. Problems caused by the Product Owner were in those situations when he did not have the management's support to actually work according to Scrum [A8].

Scrum Team. According to the articles, positive aspects of Scrum, from the perspective of Development Teams, were better team spirit, focus and better communication [A6], [A13], [A8]. Developers were also more satisfied working with Scrum than older development models like Waterfall [A13], [A8]. It was also noted that sometimes communication was not at a level that could satisfy Scrum needs, or that team spirit was not good [A6], [A3], [A10]. Also team members who did not understand Scrum could not work effectively [A10], [A0]. According one article it also required a lot of discipline to attend all the meetings and meet the Sprint goal [A4]. Issues arose when people were in different locations and from different cultures [A10]. Working with non-agile teams was also found to be demotivating [A13].

According to the VersionOne survey, the top 5 barriers for agile adoption were inability to change organizational culture 55%, general organizational resistance 42%, pre-existing rigid/waterfall framework 40%, not enough personnel with the necessary agile experience 39% and management support 38%. Benefits by VersionOne top 5 were adaptability 87%, increased team productivity 85%, better visibility 84%, increased team morale/motivation 81% and better delivery predictability 81%. [A14]

4.3 Modified Scrum according to the articles

It is clear Scrum practice benefits were noticed but Scrum was not always used by the book; sometimes it was employed only partly or combined with other work methods, depending on the context and company using it. Below are some findings from articles that support this statement. Research conducted in Ireland in 2010 and 2012 noted that there was significant tailoring of the Scrum practices in many companies [A5]. Another research article combined the best parts of V-Model and Scrum, joining V-Model phase structure with SCRUM flexibility, disciplined phase transitions and project management methodology, to increase deployment speed, ensuring quality, specifications fulfillment and safety requirements. [A11] The third article argued whether or not there is a need to implement full agile methods, concluding that it is all about the people practicing it. *“Does adopting only part of Agile methods limit the full benefits that Agile is supposed to deliver? We found companies that are fully implementing Agile methods to have positive perceptions during this research. To them, nothing is bad about Agile; however, again, it depends on how people in the team are practicing it.”*[3]

Yet another research stated that there are different versions of Scrum used which have been altered from the textbook Scrum methods. “*Our findings indicate that not all Scrum principles are suitable in each context which also refer to other further studies in this field. Why not to adapt to full Scrum - interviewees explained that the meetings themselves, as well as their preparation, are too time-consuming*” [8]. Vuorinen also concludes in his thesis that there were differences in how well Scrum methods had been adopted and used by the companies in his research [A8].

4.4 Questionnaire findings

In this section, the results from the questionnaire are presented. A comparison of how findings comply with the textbook Scrum is made here, together with improvement suggestions. Since companies of the respondents were not contacted, only the team members, this thesis does not provide team- or company-specific improvement proposals. All practices are handled independently and suggest possible improvements.

There were 14 respondents to the questionnaire, though one of the respondents was not sure if his team really use Scrum, since it only used parts of it. This respondent’s answers will still be included here, since they shed light on the question of which Scrum practices are actually seen as valuable.

4.4.1 General overview of the respondents

There were a total of 14 responses to the questionnaire; for overview of the companies (see *Table 3*). Here is presented some general data from the respondents: 12 different companies; roles 4 SMs, 2 POs and 8 Software Developers (see *Figure 4*); countries: Germany, Denmark, France, Finland, Sweden, Japan and Syria; Scrum Experience on average 3.75 years. The distribution is in *Figure 5*. Two of the answers were not answered in a way to comprehend on numerical values, so they were marked not applicable (NA).

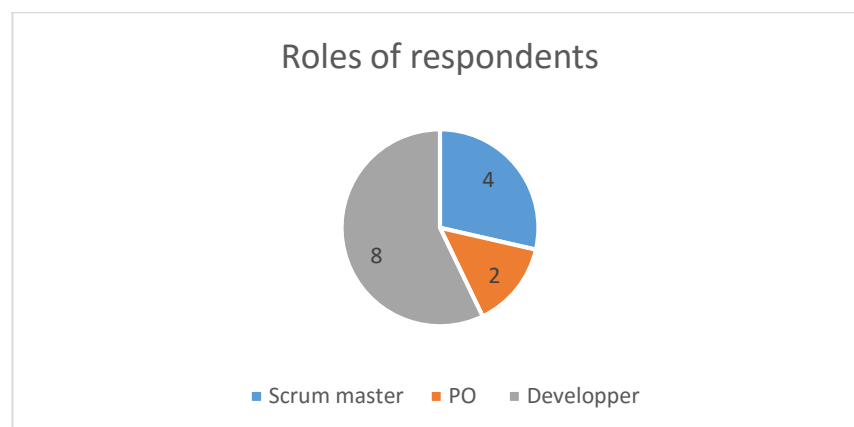


Figure 4 Overview of respondents’ role in Scrum Team

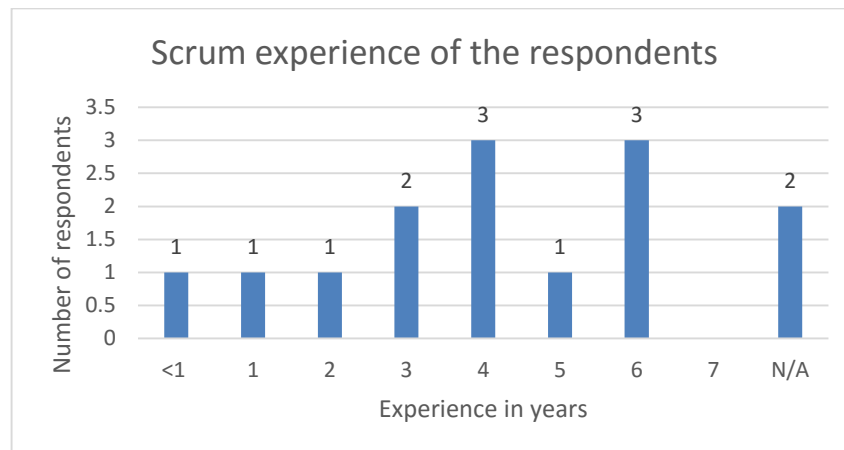


Figure 5 Respondents' Scrum experience

Table 3 Respondent general overview

Company	Answers	Role	Scrum experience	Local/distributed	Team size
A	1	SM	3 years	Local	6
B	2	DEV	Both 4 years	Both distributed	8 & 10
C	1	DEV	3 years	Local	7
D	1	SM	6 years	Local	6
E	1	SM	NA	Distributed	9
F	1	PO	6 years	Distributed	5
G	2	PO & DEV	Both 4 years	Both Local	12
H	1	DEV	6 years	Local	8
I	1	DEV	1 year	Local	10
J	1	DEV	2 years	Local	6
K	1	SM	5 years	Distributed	8
L	1	DEV	<1 year	Local	5

4.4.2 Scrum Team

Team size. Team size of the respondents' teams varied from 5 to 12 people (see *Figure 6*). According to Schwabers book the ideal team size is 7 plus/minus 2 [31]. Most of the responding teams fall into this category. There were group sizes of 10 people in 3 companies and one team of 12 people; these were bigger than the recommendation by Schwaber. When teams are big then there is also more complexity; much more effort is needed for the communication and handling of the meetings. If the teams are big, the recommendation would be to split the teams into 2 Scrum Teams to decrease complexity. This way meeting time, for example in the Daily Scrum, will decrease and more time will be left for coding work.

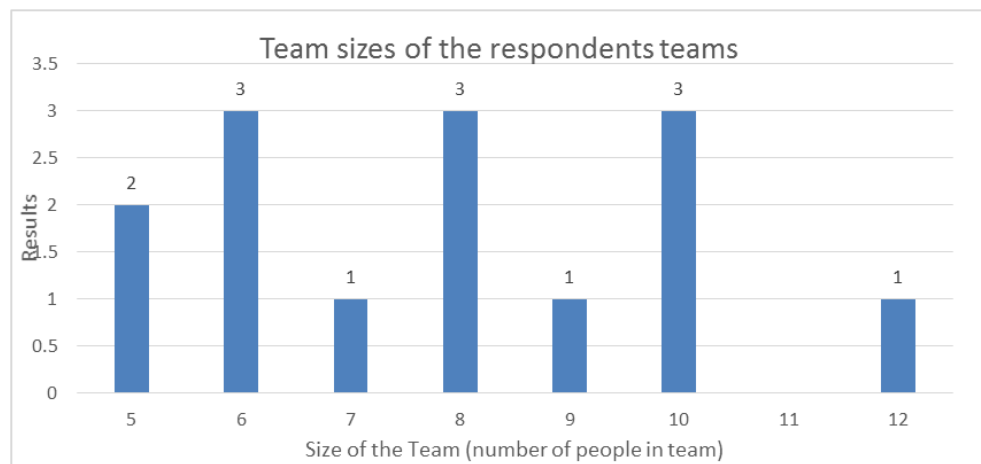


Figure 6 Size of Scrum Teams

Team location. The split between team locations was that 9 out of 14 were local and the rest were distributed (see *Figure 7*). It seems more and more distributed teams are starting to do Scrum [26], [45]. There are aspects of Scrum that can help the usual issues seen in distributed teams, like daily meetings for info and knowledge share, dedicated roles and clear responsibilities, and visibility and focus [11]. So it is advised to take special care with distributed team setup, to include all the team members in needed meetings and trainings. Scrum Master and Product Owner role is more important, they should keep up the communication and cooperation. One respondent mentioned in one answer that they have conference phones and chats to have the non-local team members updated and attending to Daily Scrum.

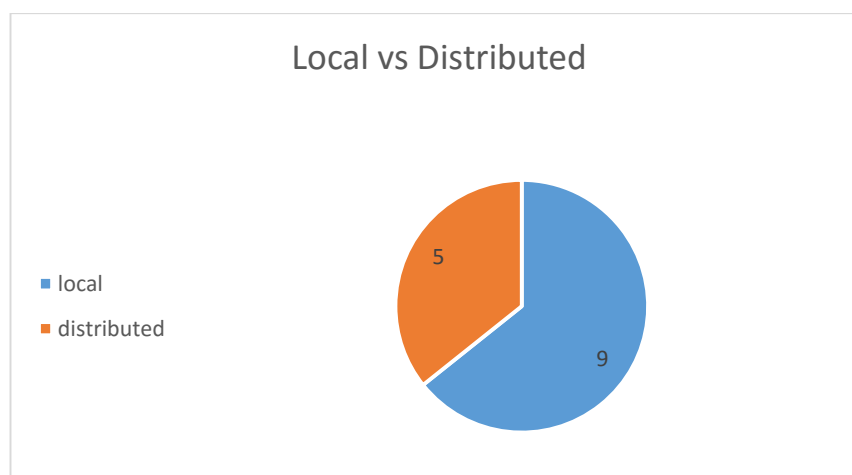


Figure 7 Team location local vs distributed

To the question what does Scrum Master do in your team, following answers were given:

- There is no Scrum Master.
- Ensure that we plan and progress, that we fulfill DoD and evaluate the Sprint and plan a new one.
- Mainly facilitates communication.
- SM duties and also development / testing work.
- Facilitate meetings.
- He is the team lead and not always present at standup meetings.
- Co-ordinate, motivate, focus and keep it together.
- Equals to line manager in our "not Scrum" team.
- Organizes meetings, keeps track of velocity.
- Organizing the meetings, taking notes of the complaints and delivering feedbacks.
- Role has decreased with maturity.
- Facilitates collaboration, meeting.
- it is the guarantor of the ceremonial.
- Coordinates work between team members and puts short-term goals.

The following question asked respondents if they felt that the SM helps to achieve the Sprint goal, whereby half of the respondents felt he did, and the rest of the respondents felt he is contributing only somewhat; one said the SM is only contributing to the team process not the Sprint goal (see *Figure 8*). Of the ones answering yes, one mentioned that the SM is an important person for keeping the team attending the meetings, so he is a motivator to keep up the Scrum Events.

We can conclude that half of the teams seem to be happy with the SM actions and the other half would find some level of improvement welcome. One of the respondents mentioned *"We are missing a Scrum Master that are enforcing good Scrum practices"*. Teams without an engaging SM should rethink the person for the role. It could be the SM is missing some Scrum knowledge or is not motivated for the role. This issue should be taken up in the Retrospective meeting by the team. This is of course requires an open and honest discussion by the team, and way to present the problem respectfully.



Figure 8 Scrum Master helping to achieve Sprint goal

Product Owner. The Product Owner is the person to control the Product Backlog [31]. There was no direct question in the final version of the questionnaire to ask if respondents have a Product Owner on the team, but based on the answers there were at least 10 teams which had one. 4 teams are noted as unknown.

4.4.3 Scrum artifacts

Product Backlog. Requirements of the product are listed in the Product Backlog, which evolves and changes as requirements get more detailed during the project [31]. From respondents' answers it is clear that 12 of the teams were making the estimates for the backlog. This is how it should be done according to Schwaber [31]. None of the respondents stated that estimates would not be asked of the team. From this we can conclude this aspect of making estimates seems to be in good shape in the teams taking part of the questionnaire.

Nine of the teams were using Planning Poker to make estimates. Five teams mentioned estimates are pretty accurate and in contrast 5 teams answered that there are still a lot of estimation errors happening (see *Figure 9*). The teams not using Planning Poker could try and see if using it would decrease estimation errors. The high amount of errors in estimating can also be due to how long the team has been working together, if they still don't

know their velocity and the technology domain. Also the type of work they are doing can affect estimates; if teams are introducing new technologies all the time it is not that easy to estimate since basic knowledge is still lacking. Review one responder's comment relating to estimation errors: *"We are now working on web based products, and most of the team is still learning many of the needed technologies. We realize better ways to do things, and sometimes have to refactor quite a bit of code because of this "*

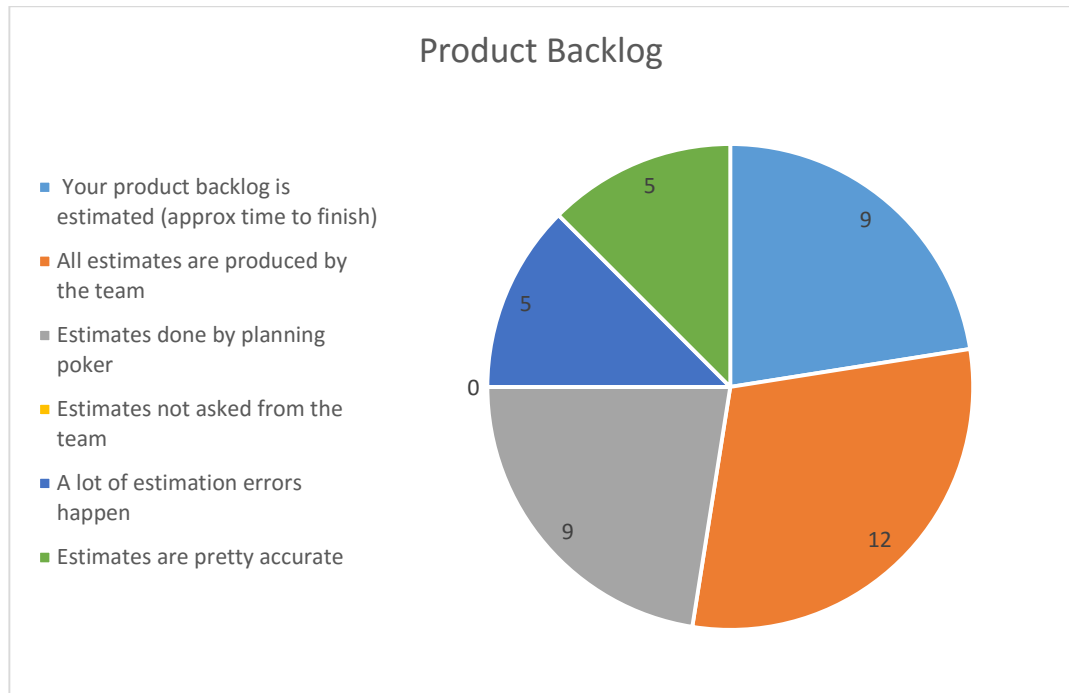


Figure 9 Product Backlog related findings

Burndown Chart. 8 teams were using the Burndown Chart, 6 did not. Of the 8 teams that used it, in 6 teams the team members updated the Burndown Chart. *Figure 10* shows the results for what is tracked by the Burndown Chart: in 6 cases Stories done, in 4, tasks, and also in 4, hours and days.

The most agile practice would be to track only tasks and stories done, and not just track hours since they do not tell how much actual progress has been done [42]. More visibility and transparency could be achieved to start moving to track tasks or stories done.

Six teams which were not using Burndown Chart at all could benefit by introducing this to the team. It can provide real time information to different stakeholders, and in seeing the actual progress, some of the enquiries could be avoided, giving more time and peace for coding work; management could have more confidence on the progress of the Sprint and project in general.

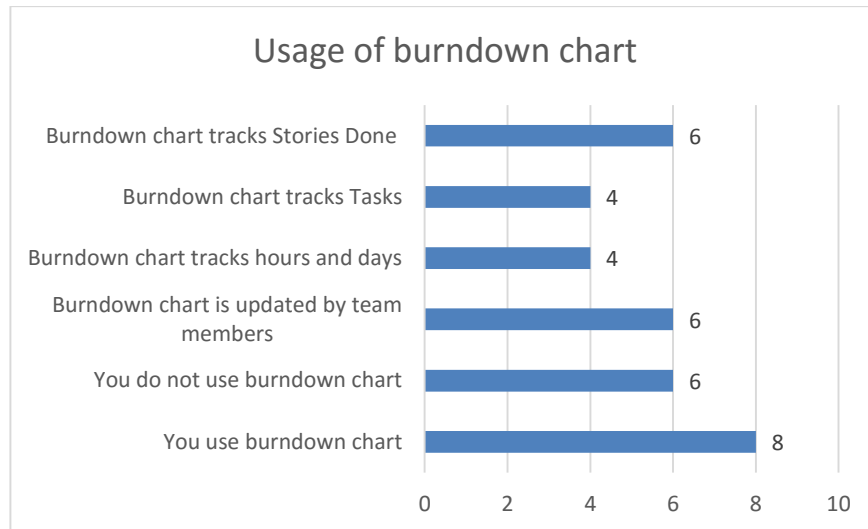


Figure 10 Usage of Burndown Chart

Tasks and interruptions. Most of the teams were interrupted based on the respondent's answers: in 10 teams out of 14, only 4 teams were left to work in peace for the Sprint. Only one team mentioned they were interrupted many times, *Figure 11*. The Sprint should be time for the team to focus on fulfilling the Sprint goal and have no changes to the Sprint Backlog; the only meetings should be the Planning, Review and Retrospective and Daily Scrum, where status is shared every day [36]. Based on this questionnaire finding, there could be room for improvement in stopping interruptions in order to get a more productive team. This questionnaire did not ask for the sources of interruption. In one team the organization was not otherwise working in an agile way, as the respondent comments "*Others in the organization are disrupting the team*". These teams should bring up these interruptions in the Retrospective meeting and try to bring down their amount. If the team is working with other teams that do not practice Scrum, it is likely they do not respect the Scrum practices. Also if some of the other stakeholders do not work in an agile way they can also cause interruptions. Here the Scrum Master has the important role of blocking the interruptions of the team.

Task allocation happens by choosing individually in 10 teams and by assigning by SM or someone else in 7 teams. From this data we can assume in some teams both ways are being used. According to Scrum, teams should be self-organizing and tasks chosen individually [36]. It seems some of the teams are still being managed from top down PO or SM telling team members who should do what; possibly team members don't take responsibility for this and the PO and SM are forced to assign tasks. Team members should be disciplined to follow the Scrum process. Also the Scrum Master and Product Owner need to give the freedom to team members to choose the tasks and not to manage too much. Getting the team to organize itself is one important part of Scrum, as are taking responsibility and ownership. Unmotivated teams should focus on improving the team spirit and teamwork, and emphasize learning. It can be argued that not all people are suitable for working with Scrum, due to the high level of discipline it requires.

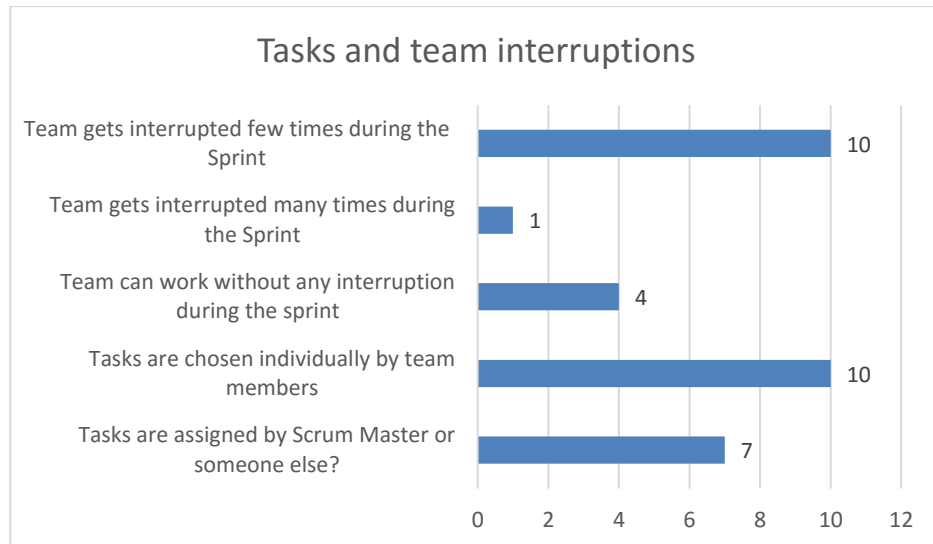


Figure 11 Tasks and team interruptions

Velocity. 9 teams responded as knowing their velocity, 2 did not know and one team had just started so it did not know yet; 2 respondents skipped the question. Most of the teams who knew their velocity measured it with story points. 2 of the teams mentioned they measure with tasks done in a week, while another counts tasks done in a Sprint. Only in knowing the velocity of the team is it possible to estimate the Product Backlog and give estimate of the project/feature deliveries [31]. Teams who do not know their velocity could improve overall Scrum process by measuring their velocity, making it possible to do better estimates to deliver releases and improve customer satisfaction.

Requirements. Scrum requirement management differs a lot from traditional requirement management. Requirements do not need to be ready in the beginning of project. Requirements are often in a form of user story, which become more detailed during the development. [31]

Here are some comments related to requirements from the respondents:

- Marketing write UX stories, approved by SOFTWARE and UX.
- PO writes user stories.
- It much more flexible than "traditional" requirement handling, but PrM/PO needs to be careful that the workload in Sprint/Increment stays in balance when the requirements change. We also use EPICS.
- Requirements should be reflected in the backlog.
- We use epic, stories and task in JIRA to drive our requirements.
- Defining requirements is a continuous process, first it needs PO to come up with initial specification, then a developer tries to convert it into some form of SBE(spec by examples) which is not exhaustive but covers most of the user scenarios. Changes are always welcome during implementation.
- There is a mapping with user (technical) stories and requirements.

Two teams mentioned they have requirements ready already when the project starts; this is more of a traditional Software Development practice, and misses out on Scrum's flexibility to adjust to and accept new requirements. These teams should look more into making it possible to have flexibility in their backlog. The Product Owner is in central role here, and has the authority to control the backlog. Sometimes it might be difficult if the team is working with teams that do not use Scrum. They require input for example on interfaces etc. early on to be able to do their work. 7 teams had changing requirements during the project and 10 teams were using user stories. User stories are good way to have requirements done from a user perspective.

Definition of Done (DoD). To be able to mark task done, there should be criteria what is needed to be done before a task or a story can be marked as done. This can vary according to teams, but everyone should have a clear idea for their team. [36]

There were 13 answers, and one skipped the question; the Definitions of Done according to each respondent's team are written below. DoD changes according to team and some have very specific criteria and some higher level description. Higher level descriptions could be good to develop in more detailed one, to be sure all team members are in the same page and there is no discussion at the end of the Sprint of the state of tasks. Below presented all DoD definitions:

- Do I have to describe that in one sentence? :-).
- Code compiling, unit test passing, code reviewed and tested.
- Code pushed to GIT, reviewed and tested.
- - Time spent is updated - remaining estimate = 0 - applicable reviews completed (for example code review, specification review) - code is pre-release tested (i.e. it's not committed to the main development branch before the tester has given the green light.
- 1. Code review without errors using a pull request; 2. User acceptance criteria has been tested ok on QA Site; 3. All test cases with test data is run on PO defined subset of devices; 4. Bugs reported by in Sprint testing are fixed; 5. Items done correct according to "customer in question checklist"; 6. Test Automated Reviewed and tested.
- Product releases, verified and packaging completed.
- We write it in each User Story.
- Once the Sprint is finished, everything else goes in Sprint Review for new ps.
- Code delivered and unit tested.
- Code review, Unit tests, working functionality demo to Product team.
- It depends of the story and Sprint, we have a specific check list for different cases.
- Goal achieved and approved.

4.4.4 Scrum events

Sprint. Schwaber specifies in his book that the Sprint is a fixed period of time when a team works to achieve the Sprint goal [31]. All 14 respondents answered the question. In one team they do not really have Sprint but only quarterly targets.

Eleven teams had fixed length for Sprint, 2 did not have fixed length, one NA. Distribution on Sprint lengths can be seen in *Figure 12*. 2 weeks was the most common one, 8 teams were using it, and then 3-4 weeks was second with 2 responses. One respondent mentioned that Sprint length is fixed by the global project, so the duration is determined for all Scrum Teams same in the project. One person answered not applicable (NA).

Most of the teams are doing as is should be according to textbook Scrum. 2 teams had changing length according to project. One team had only quarterly target, this team could improve having shorter iterations and show the Increment to customer more frequent to get confirmation the work is what customer needs. This could prevent work being done in vain and corrections could be made more frequently.

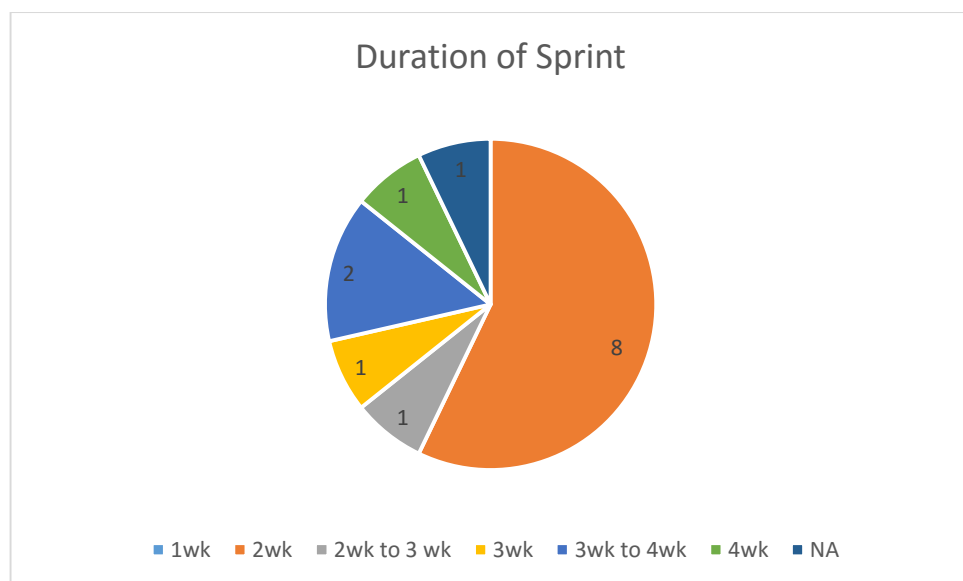


Figure 12 Sprint duration

Daily Scrum. Daily Scrum is the part of Scrum everyone was practicing, and it happens daily with the whole team. Duration is on average 15 minutes. In *Figure 13* distribution of Daily Scrum duration is presented. Everyone seemed to follow also the same agenda: everyone shares what have they done yesterday, what they will do today and if there are any impediments.

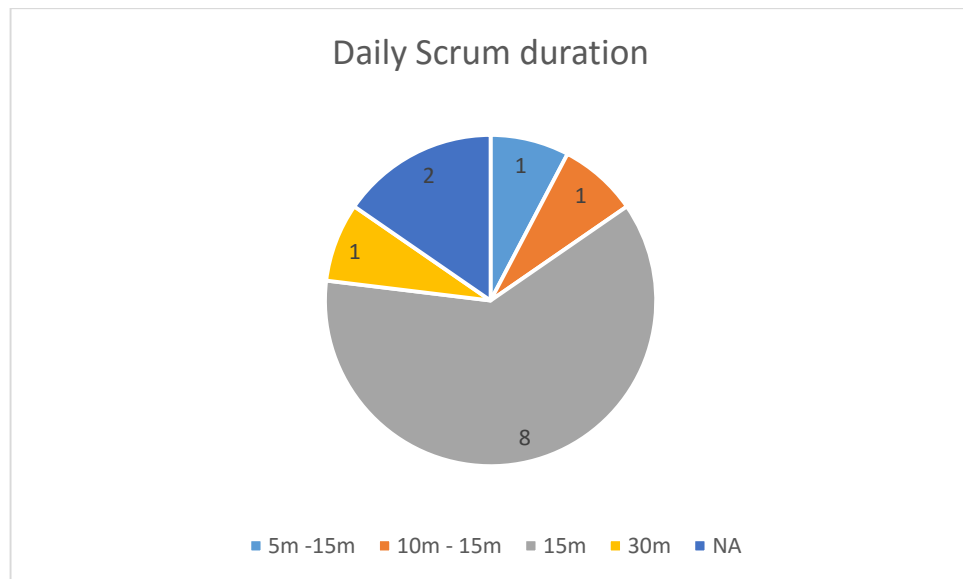


Figure 13 Duration of Daily Scrum

Here is presented perceptions of the Daily Scrum meeting by the respondents. Marked positive and negative sides with plus/minus (+/ -):

- + Motivating.
- + Distributed team attend with screen and desk phone for Indian & Chinese team; +Everybody knows about everything; + sometimes a person has a 'key' for progress (special knowledge); - time away from coding.
- + Set clear goal for the day.
- + Regular visibility to team's progress; - Sometimes goes too technical, in which case the meeting should be continued with smaller group.
- - bad side is lack of focus.
- + Good way of getting an idea of what the others are working on; - Sometimes we get too much into details.
- + Everyone knows where each other are, anyone who needs support.
- + Quick view what people have done. If someone needs help. Information sharing.
- - If it gets too long it becomes annoying.
- + Real time escalation; - keeps high pressure on the team (bad side).
- + Keeps everyone posted; - Sometimes nothing new to discuss.

Most of the teams had 15 minutes duration for the Daily Scrum, which the Scrum text-book recommends [31]. One team's meetings had 30 minutes duration; this was a team with 8 people in a distributed setup, which could cause the longer duration since there are more things to share with different locations when missing other daily encounters. However, it was commented by this team member that the meetings sometimes get too detailed. Detailed technical discussions among involved members can take place after the Daily Scrum. This way others save valuable time and the meeting does not feel too long. A 15-minute meeting per day equals 1 hour 15 minutes per week, while a 30-minute

meeting equals 2 hours and half each week. The latter is already taking a considerable time away from coding, so it should be considered to shorten the meeting to 15 minutes. Finally it is worth noting that the Daily Scrum seemed to be one of the most popular practices, and all the teams used it.

Sprint Planning meeting. 13 out of 14 teams used the Sprint Planning meeting; duration of the meeting varied between half an hour to one day between the different teams. Most common was to have a meeting lasting from 1 to 2 hours. One team had a half a day meeting and another team a full day. One team had split the meeting into 2 parts: the first with the PO to decide what is in the Sprint, in the second meeting team decides how and who does the Sprint tasks. According to Schwaber there should be 2 meetings, the first one with the PO and management to decide what functionality is built during the next Sprint. In the second meeting the team should decide how they will actually do this and make a product Increment. [31] Only one team among the respondents followed this approach.

The customer was not at the meeting for most cases: a few respondents said the PO was representing the customer, and some said that the internal customer was attending. Only one respondent stated that the customer was present in the meeting. Customer involvement is one of the basic Scrum principles. By having the customer in the meeting, many of the teams could gain more ideas and feedback from their work directly from the customer.

In *Figure 14*, it can be seen who decides what will be in the Sprint Backlog for the next Sprint. In 5 cases it was the team, in 2 others, the team together with the PO, but in the rest of the cases it was the PO or SM deciding the tasks for next Sprint. Scrum emphasizes self-organizing teams [31]; self-decision should also include task decisions. Only then can the team can be committed to the Sprint goal and know it can deliver what it promises. The PO, management and customer should be there to guide what functionality is wanted from the next Sprint; the team itself should agree on the tasks taken into the meeting. This is an improvement point for many teams.

Here are perceptions of the meeting. Positive and negative aspects are marked with a plus/minus (+/ -):

- + In SPM2 (sprint planning meeting 2) the team without PO will define the tasks needed for each US (user story).
- + Good overview and perfect prioritization of what's most relevant in next Sprint, Everybody gets aware of all projects and can better up- or down-prioritize tasks; - time away from coding (usually interrupts comment - *not now..*).
- + Gives clear target for next weeks; - keep the meeting within the time limit.
- + Everyone should understand the Sprint contents after this meeting; - not everyone are as focused as they should be.
- - Tends to be too long.
- + Knowing what each team member is planning to work on; - Not all members are prepared for the meeting.

- + Team agrees on what to do; - not all clarity is resolved due to dependency on knowledge area.
- - Not always well prepared, stories not always ready.
- + Provides a clear view of the load of work; - Sometimes it is prone to unnecessary waste of time.

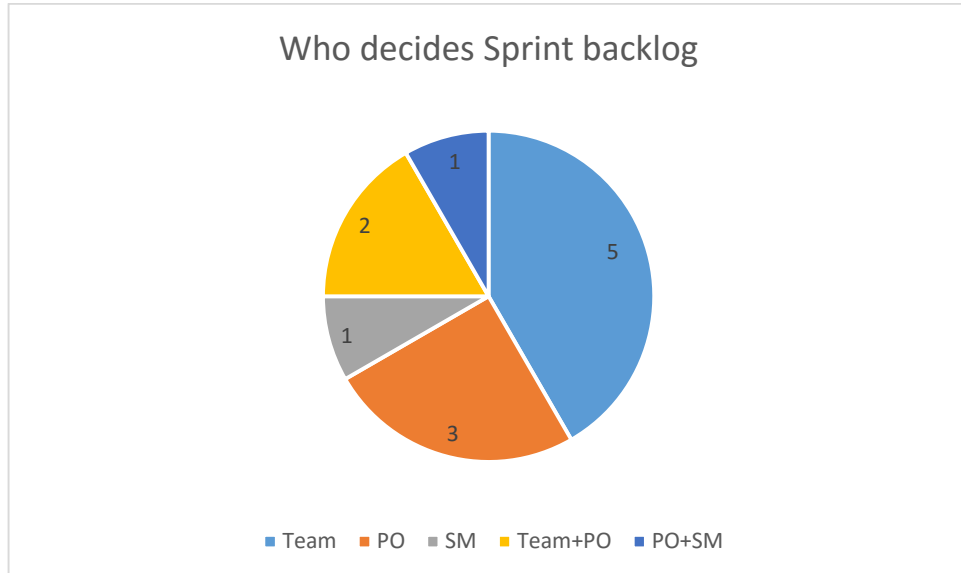


Figure 14 Sprint Backlog contents

Sprint Review meetings. 3 out of 14 teams said they did not have a Review meeting, 11 had one. Duration was mostly between 1 and 2 hours. Attendance at the meeting was in 10 cases the whole team, and 1 team had following attendance: PO, Test Manager, Quality Assurance, Release manager and customer. Asked separately later if the customer was present, 6 out of 11 teams responded in the affirmative. For those teams who did not have the customer present, future practice could be improved through including him; customer presence; can give valuable feedback to the team and ideas for next Sprint.

The Review meeting presents what has been done during the Sprint: stories done, tasks done, and the actual Increment. In one case there was comment related on what is presented “*Discussion of tasks are completed or should go to next Sprint*”. This points out that the definition of done for the tasks has not been communicated well, if the DoD needs to be discussed in Review meeting. This is a good opportunity to shape up the common understanding by agreeing on a DoD so that an Increment is produced accordingly. One of the respondents commented negatively that there is not much difference between this meeting and the Daily Scrum meeting. In this case, the meeting is not really used for what it is meant; the main objective should be to show the actual working piece of software to the customer. The Daily Scrum is just to track tasks and impediments. For this team, the meeting scope could be adjusted to fit the Scrum textbook approach to get feedback and ideas from the customer.

Here are perceptions of the meeting. Positive and negative aspects are marked with a plus/minus (+/-):

- + Confirmation of work through customer or customer representative is important for the team.
- + You are forced to argue if closing or continuing tasks; - time away from coding.
- + Good possibility to adjust to next Sprint; - so far we had no real problems.
- + Wraps up the Sprint nicely.
- + Knowing what each member did in past Sprint; - not much different from our 'stand-up' (Daily Scrum).
- + Team exposure; - real time feedback.
- + Feedback to all team members about their work.

Sprint Retrospective. 12 out of 14 teams kept the Retrospective meeting, 2 out of those 12 combined it with other meetings, one with Review and one with planning. All respondents' teams had the whole team present. The usual duration of the meeting was one hour, nonetheless, one team only had 10 minutes. Agenda in the meeting according to respondents were similar: to present what was working and what didn't. One of the respondents mentioned they used templates "*There are a few templates which has been used by teams (for example "6 think-ing hats" [44])*" another said they vote on things to improve in the next Sprint; "*What went well, what could be done better in the coming Sprint, lesson learnt and team votes on what they want to improve in the coming Sprint.*"

The main idea is to learn from the last Sprint and act to improve the working process for the next one. The important thing is to have actions based on information received in the meeting, otherwise the team might feel it is not worth it, a conclusion reflected by one respondent "*That agreed actions are followed-up, otherwise the retro kind of loses it's meaning*". Sometimes respondents felt forced to come up with some improvements for each meeting: "*Sometimes people feel "forced" to make contributions. I personally sometimes cannot find anything particularly good or bad about a Sprint, but there is sometimes this feeling of being "forced to say something", also in the previous company I worked for*" There should be a positive attitude towards the meeting; therefore, in problem cases the Scrum Master should seek ways to make the meeting work better. Also, respondents should say in the Retrospective meeting that the meeting itself needs improvement. Nothing can be improved if it is not brought to the attention of others in the team and the SM. It all comes down to to the point that the team should feel comfortable with sharing their opinions and the Scrum Master should be doing his work properly.

Here are the rest of the perceptions of the Retrospective meeting. Positive and negative aspects are marked with a plus/minus (+/-):

- + Team is feeling supported.
- + We discuss things not constructive and try to make actions for them; - is to not to say the same things.

- + People say more easily what bothering them than normally. A good template / agenda helps a lot.
- - Tends to be too long +All good! This is the chance for team to make continuous improvement.
- + Need to insure clear actions are taken out of this meeting.
- + Corrects the way things are done; - Not necessary every Sprint.

4.4.5 Engineering practices used

Questionnaire results show the most popular engineering practices used in companies in *Figure 15*; peer review is the most commonly used engineering practice. Respondents also emphasized teamwork and achieving together, reducing the amount of errors. Other practices used widely were Weekly builds (9), continuous integration (9) and unit testing (8). All these actions contribute to the fact that better quality code is being written and delivered, tested and integrated with the baseline in short circles. Story points were also used in 9 teams.

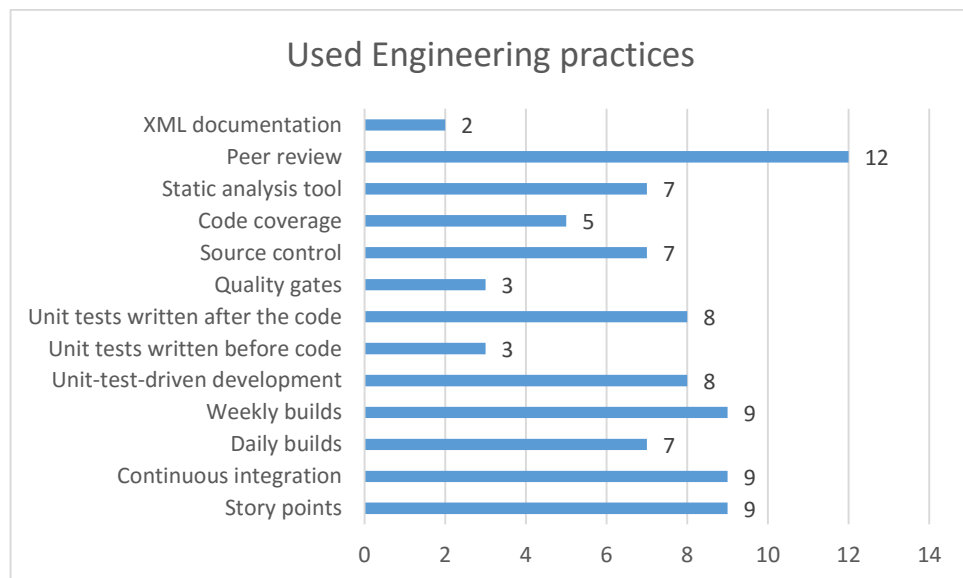


Figure 15 Engineering practices used

4.4.6 Questionnaire findings on tools usage

Communication. Email, chats, and Messenger were the most common communication tools. Email was the most popular tool and second came JIRA, Skype and chats, all with 2 answers (see *Figure 16*) [22], [33]. Each team communication tools seemed to be changing according to the fact what were the company provided tools. Other Agile project management tool mentioned here were HP-ALM and Slack, kind of an irc-channel for IceScrum agile project management tool [19], [20], [34].

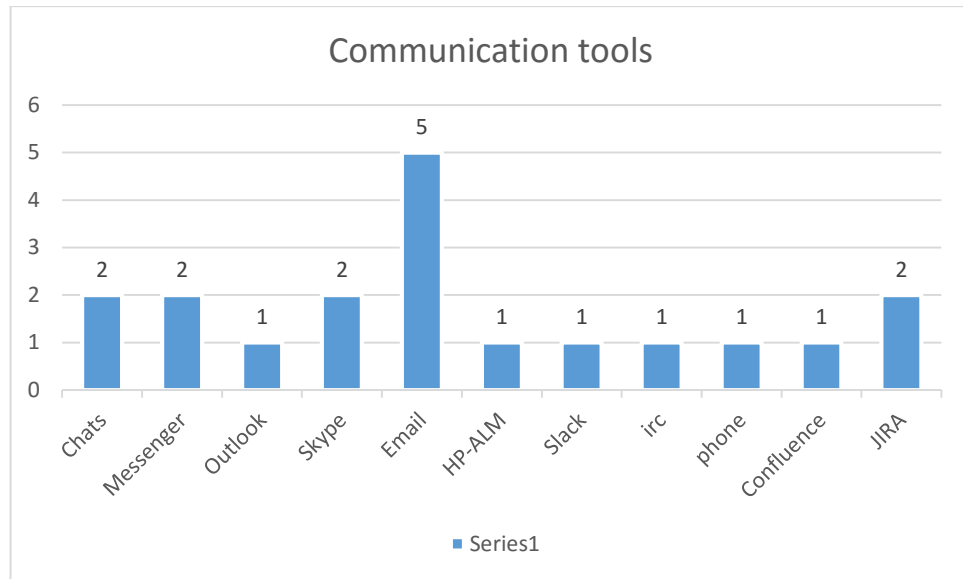


Figure 16 Communication tools

Version control. GIT is clearly most popular tool, then second SVN (Apache Subversion) [4], [14]. Responses amount was 16 answers, this is because some companies were using more than one tool depending on the project, or they were in transition to another tool.

Testing. Testing has been split into many different tools (see *Figure 17*); this is because there were also many different languages used in companies and areas of business functions. This questionnaire did not specify which languages are used by respondents and in which business area they work, so a further detailed comparison is not possible.

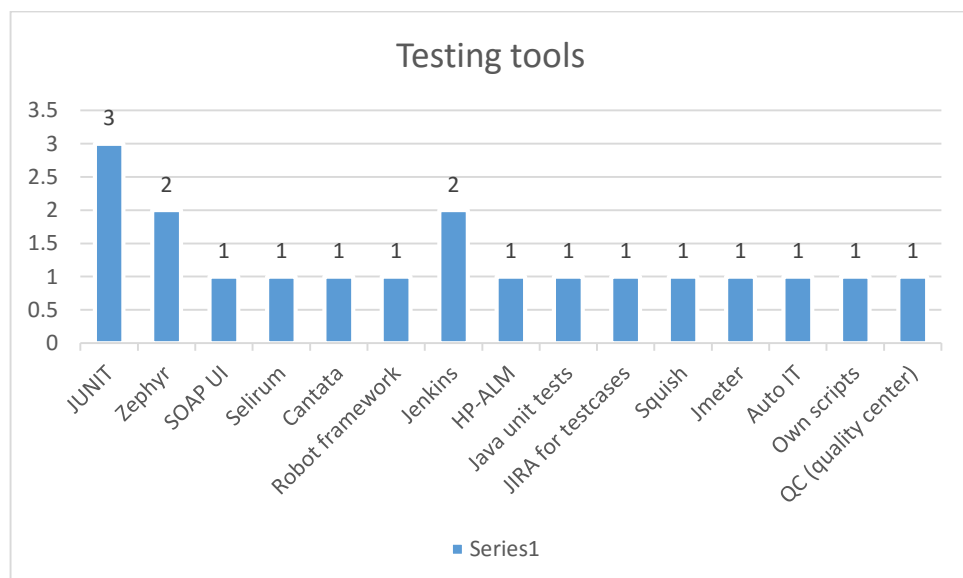


Figure 17 Testing tools

Agile project management. For agile project management JIRA seemed to be the most popular tool [22]. There were a total of 12 responses to this question. All the mentioned tools are in *Figure 18*.

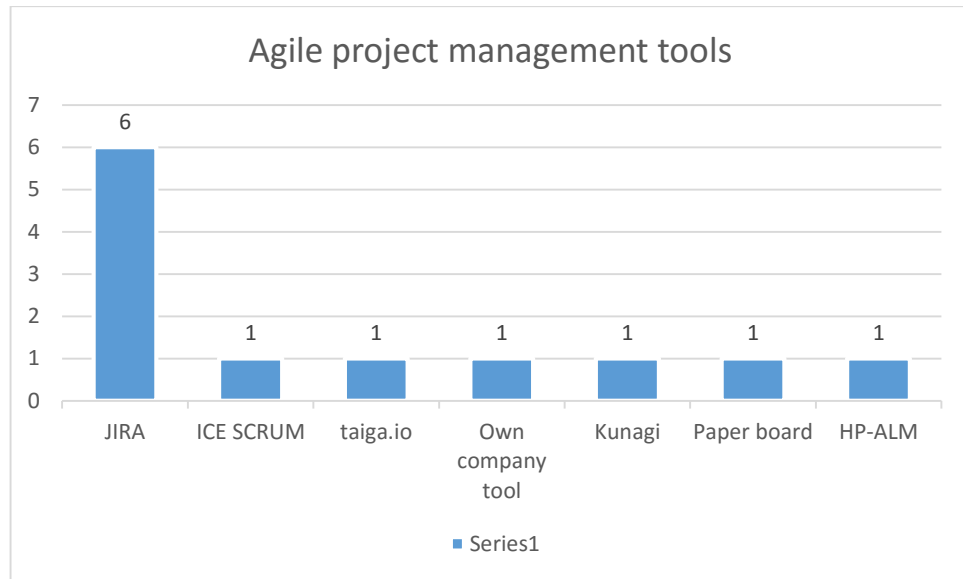


Figure 18 Agile project management tools

Other tools mentioned were Confluence for requirements and meeting minutes, JAMA for requirement handling, and JIRA used for error reports [21].

4.4.7 Testing

In the questionnaire question concerning testing, *Figure 19* charts the following results. Unit tests were the most common tests, but only 8 out of 14 teams used them. Only in 6 teams was testing part of the code delivery. Only 4 teams smoke tested integrated code. That is one practice whereby many of the teams could shape up the code quality by doing more testing in order to deliver working and tested code.

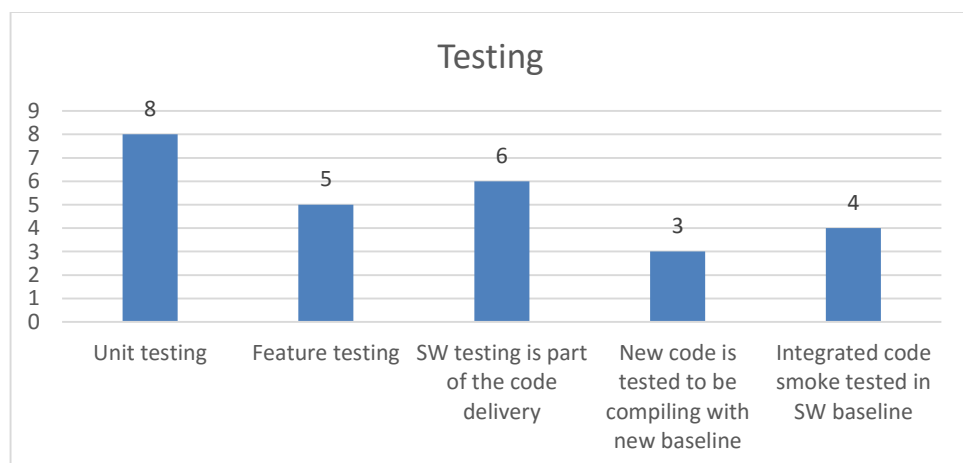


Figure 19 Different ways of testing used in respondent's companies

4.4.8 Positive and negative aspects of Scrum

In the questionnaire there were 6 open questions about Scrum which gave more insight into the perceptions of using Scrum. All answers are in *Table 4*.

With respect to the question **What is the general feeling towards Scrum in your team?** All answers were positive, so it seemed that people liked Scrum. This is in line with the findings in article studies: developers are more satisfied when working with Scrum. One of the respondents commented following:

“It is good. Once the teams are self-sufficient, they feel empowered to make daily decision towards success of the project”

Next question **What kind of feedback you get about using Scrum from your customer?** Most answers said they have good feedback, for example that it *is dynamic, adaptive, greater sense of control*. But some issues also arose regarding the delivery time:

“Generally good but the expectations from customer is still based on a set time. The whole industry is adapting to the Scrum ways, so everyone understand but we a long way from it becoming the norm.”

Also, a few answers mentioned that the customer did not know about Scrum or the customer did not approve. In these situations, probably the Scrum process and its benefits had not been communicated well to the customer. Schwaber writes *“Establishing an open relationship with the customer is the most important aspect of Scrum; Scrum makes everything visible”* [31]. Of course in some cases, the customer might think there is too much time-consuming communication, as Vuorinen states in his thesis [42]. At any rate, based on the answers here and the previous data from the customer attendance in meetings in *Section 4.4.4*, there is room for improvement in many companies to have more cooperation with the customer and to keep the customer involved with the project.

The third question focused on the learning aspect **What are the leanings in your team when you have been using Scrum?** For the learning questions there were many different points of view, some more precise like top- down work breakdown, defining stories and others comprehend the very core of Scrum; learn from past and evolve as a team:

“We deliver what we plan, we improve continuously and learn from our mistakes quickly. We have become more confident as a team and the whole team feels the ownership”.

Learning is the core of Scrum, and the team should continuously inspect and adapt its working methods.

The next question asked about the good aspects: **What you think are positive sides of Scrum?** Answers followed mostly same principles mentioned before in this thesis when

discussing the research articles in *Section 4.2*: its adaptability and flexibility, teamwork, focus et cetera:

“Agility, adaptability and possibility to course correct continuously with the right input from stakeholders / customers”

The fifth question asks what needs to be improved **What does not work with Scrum?** Respondents mention that working with Scrum in non-agile company is causing issues, also working with teams having deliveries from team doing waterfall approach causes hiccups. Long term planning and some tools constraints caused also negative feelings.

Few comments from respondents:

“Being dependent on deliveries from waterfall teams” and *“Hard to plan tasks good enough long time ahead - often we only plan for near future (deepening on the project of course)”*.

Issues arising from the bigger picture such as organization culture and non-agile surroundings. Also the fact planning long term is different from traditional development methodologies.

The last question was **What are the challenges with Scrum?** There was a large variety of answers here: long term planning, management support and understanding, transparency between client and the team, estimation of tasks. These answers are very much in line with the findings from articles in *Section 4.2*. Here few comments relating to challenges:

“Relies on good collaboration within the team, Needs patience and commitment from management as it forces them to work differently”

“Not really understanding all the technical de-tails at the moment of estimation. and too positive thinking”

Table 4 Perceptions of Scrum

Question	Respondents answers
What is the general feeling towards Scrum in your team?	<ul style="list-style-type: none"> • Positive • Good, That it is beneficial - now everybody knows everything and we can spread tasks easily. • Usually we reach our goal and we stick in our promises. • Positive. • Seems like yet another way of planning, we are currently very relaxed in the way we are doing Scrum.

	<ul style="list-style-type: none"> • It is good. Once the teams are self-sufficient, they feel empowered to make daily decision towards success of the project. • Looks ok for implementing new SW kind of work. Does not work well for firefighting type work. e.g. too much overhead, too slow. • Good. • Great. • Mixed. • Positive. • Good. • Convenient and straightforward to follow.
<p>What kind of feedback you get about using Scrum from you customer?</p>	<ul style="list-style-type: none"> • Positive. • Mainly quality people care as they can look at statistic. • Customer doesn't know. • Positive. • Greater sense of control. • They do not accept that we are planning in 2week steps. • Generally good but the expectations from customer is still based on a set time. The whole industry is adapting to the Scrum ways, so everyone understand but we a long way from it becoming the norm. • They like it, they understand it. • Dynamic, adaptive. • Better predictability. • Positive. • Good. • There is no such information access for the customer.
<p>What are the leanings in your team when you have been using Scrum?</p>	<ul style="list-style-type: none"> • Today everybody participate and share opinions even on areas that they normally don't know about. It helps us despite the time used - it is a lot easier to share tasks and help each other + it gives us visibility of challenges and progress + it gives us arguments why we sometimes have to reject tasks. • Defining stories with proper scope and breaking them down to manageable tasks is sometimes hard. • It should evolve all the time, so if something's not right then we'll change it. • To focus on "what works". • We are missing a Scrum Master that are enforcing good Scrum practices. • We deliver what we plan, we improve continuously and learn from our mistakes quickly. We have become more confident as a team and the whole team feels the ownership. • Good. • Unclear question. • Top down work package breakdown.

	<ul style="list-style-type: none"> • Team cohesion. • Teamwork and keeping everyone posted of the work of others. • That I provide important feedback to the other team members.
<p>What you think are positive sides of Scrum?</p>	<ul style="list-style-type: none"> • Reactivity, adaptability. • We on daily basis evaluate if somebody needs help or can help others. • good overview and feeling of progress • Burndown Chart It gives great visibility to team's progress, Teams have very good working spirit, not too much multi-tasking, People stretch if necessary. • You get to focus on your tasks. • Always making the tasks with highest priority • Agility, adaptability and possibility to course correct continuously with the right input from stakeholders / customers. • Somewhat light weight. Job rotation. Easy knowledge sharing. • Flexibility. • Perfect for proof of concepts. • Strengthen team commitment. • Early feedback, better planning, better requirements, better collaboration between team and PO/business experts. • Driving by human contact. • Short meetings - small teams - chance to fast learn from mistakes.
<p>What does not work with Scrum?</p>	<ul style="list-style-type: none"> • Hard to plan tasks good enough long time ahead - often we only plan for near future (deepening on the project of course). • Estimates: everyone sees amount of work in different way and its sometimes difficult to "normalize" them. • With Burndown Chart, At least in JIRA you cannot really change the Burndown Chart parameters once started. For example, if someone is sick for some time, it should be taken into account. • Being dependent on deliveries from waterfall teams. • Difficult handling interrupts which are not in the Sprint. • Unclearity about what is Scrum causes frustration during adaption phase. Operational projects are harder to implement with Scrum as they are time cascading. • Does not work for Research projects, when it's not clear how to implement things. • A badly managed project or badly divided in small tasks one. • Agile team in non-agile company. • Not very easy to make product quality improvements (nonfunctional). • If team or customer is in opposition.

	<ul style="list-style-type: none"> • Big critical projects
<p>What are the challenges with Scrum?</p>	<ul style="list-style-type: none"> • Still being consequent, following a long-term strategy • Estimates, we see a good progress in getting better to estimate. We also dare stating higher estimates because experience show that it first guess is typically right • Estimating still hard, we are now working on web based products, and most of the team is still learning many of the needed technologies. We realize better ways to do things, and sometimes have to refactor quite a bit of code because of this • Not really understanding all the technical details at the moment of estimation. and Too positive thinking • People do not understand Scrum • Unclarity across organization, industry and within the team. Scrum doesn't mean no planning, it means just enough planning but that leaves room to interpretation and hence communication is key towards clarity in Scrum. • Making sure that management understands it • keeping the doc updated • Line management positioning • Relies on good collaboration within the team, Needs patience and commitment from management as it forces them to work differently • Transparency between the client and the team • Harmony between team members

4.4.9 Actual outcomes seen by the teams

When asked what are the benefits teams saw using Scrum, almost all 13 respondents said transparency is better. Second with 11 responses each were teamwork is better and team is open to face and point out impediments, and team members respect each other and continuous learning is happening.

Team is committed to the Sprint goal got 9 answers. Only 3 teams seem to be free and self-dependent to meet the Sprint goal. The very core of Scrum is to have the team decide how it will reach the Sprint goal. In practice this might be difficult, as first of all it requires a lot of discipline, and second there needs to be support and understanding of Scrum on the part of the company and management to do that.

Only 4 teams seemed to have an effective way to remove impediments. The Scrum Master should step in and remove impediments for the team, however there were some teams where the role of the Scrum Master was not seen as that helpful. It could be concluded that there is possible room for productivity improvement in these teams if a new SM who can take a more active role is selected. This recalls the Scrum Master *Section 4.4.2* where

in there were teams who did not have or did not see the SM role helping the Sprint goal. In these teams the SM role should be stronger to support the team.

Three teams have the Scrum events as their only meetings; this is a very low number. According to Scrum, Planning, Review and Retrospective with the Daily Scrum are the only meetings needed; in practice there probably are more meetings needed for line management, and for the company and customer side. In *Figure 20* all results are represented.

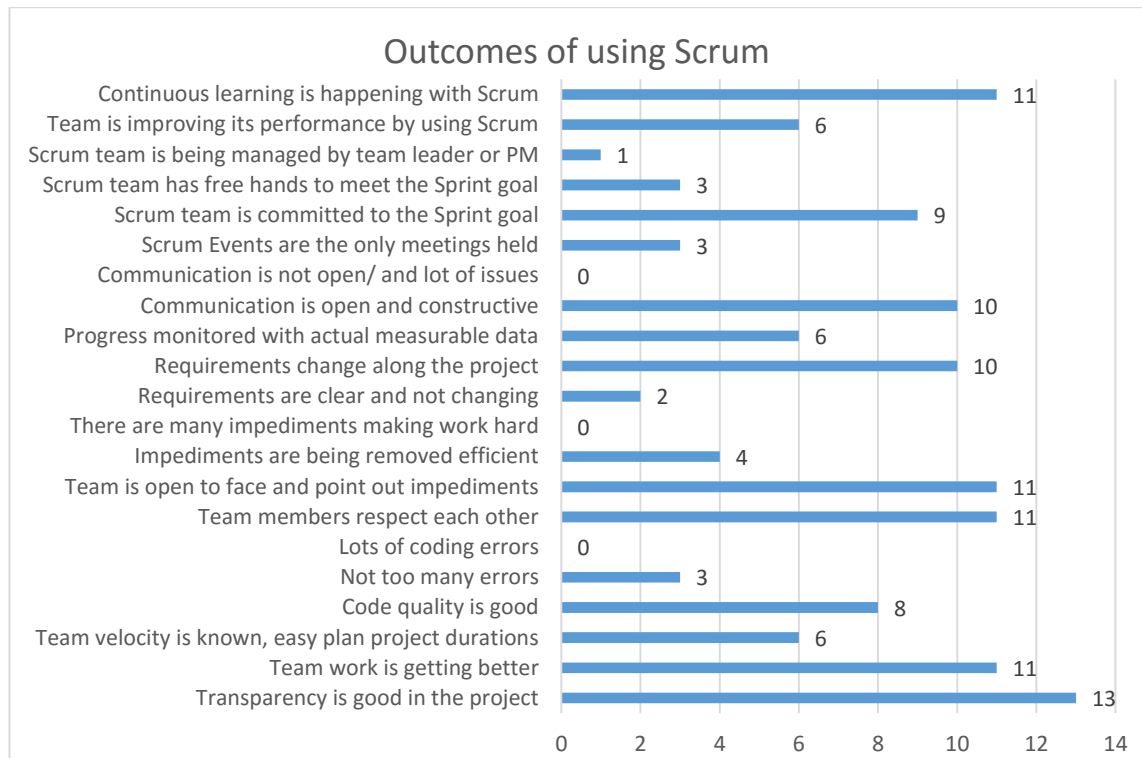


Figure 20 Outcomes of Scrum

4.5 How do questionnaire results compare to previous studies

This section compares, the questionnaire findings with the background material articles. The percentage number for articles, in *Figure 21*, is calculated from 10 articles. The Vuorinen thesis and VersionOne survey results are presented separately when applicable, hence they are excluded from percentage calculations [42], [45]. Three articles were not included in the percentage calculations, since one focused solely on tools usage and no other practices were mentioned [12]; another focused on the practices and what was good about them, but no actual usage was studied in company [11]; and, the last one surveyed the outcomes of using Scrum, but did not specify practices used [9].

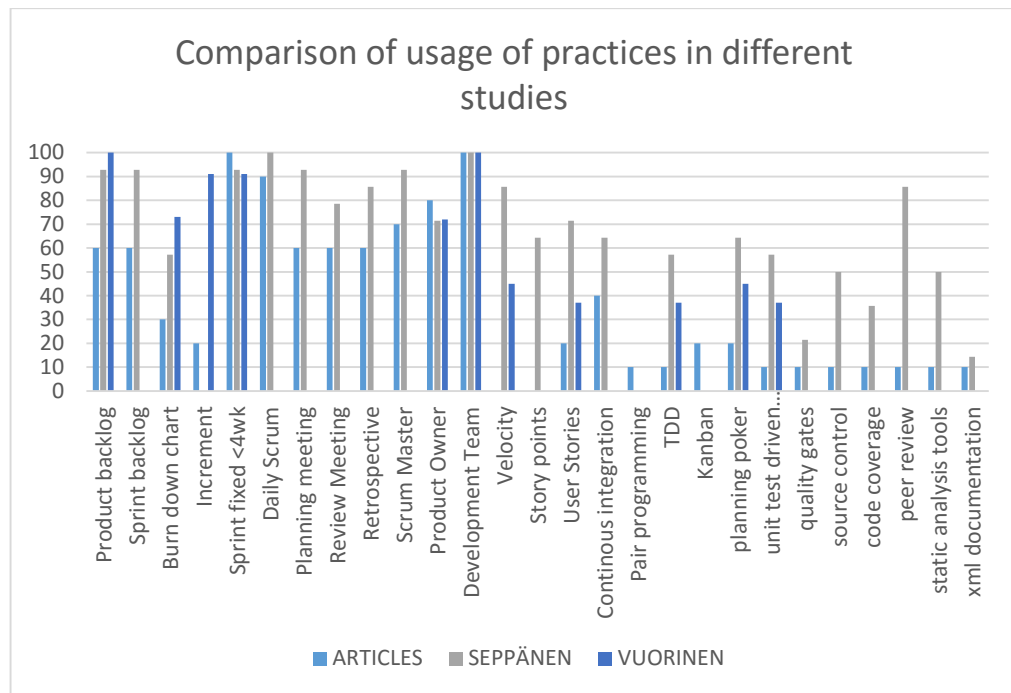


Figure 21 Comparison of practices: articles, questionnaire and Vuorinen thesis

Following percentage numbers are given in same source order in following points: *articles* (A) %, *this thesis: Seppänen* (S) %, *Vuorinen thesis* (V) %.

Product Backlog is used: Articles (A) 60% , this thesis: Seppänen (S) 93% , Vuorinen (V) 100% making it one of the most popular practice. There were high percentages on fixed length Sprint (A) 100%, (S) 93%, (V) 91%, also the Daily Scrum seemed to be one of the most popular practices used (A) 90%, (S) 100%, (V) N/A. [42], [45]

Big differences are found in the usage of Burndown Chart (A) 30%, (S) 57% and (V) 73% and VersionOne survey states 51%. This thesis result is pretty much in line with VersionOne. [45]

A discussion of velocity was not found in the articles at all, while this thesis research shows (S) 86% were measuring velocity; Vuorinen states (V) 45% uses it. The VersionOne survey states 57% use velocity. It seems the teams responding to the thesis questionnaire were above average in velocity measuring compared to Vuorinen and VersionOne. [42], [45]

The last practices in the graph are engineering practices used in companies; these were not studied at all in Vuorinen thesis, but some practices were mentioned in the Scrumbutt results, such as Test driven development (A) 10%, (S) 57% , (V) 37%, Planning Poker (A) 20%, (S) 64%,(V) 45%, and Unit test driven development (A)10%, (S) 57%, (V) 37%. Percentage comparison to articles is not that interesting since only one article researched particular engineering practices used. It is interesting to see which practices are still being mentioned in other articles too, in addition to previous ones: Kanban is in 20%

of articles, and the most popular practice was Continuous Integration with (A) 40%,(S) 64%,(V) NA. Questionnaire show peer review was used is 12 teams out of 14 (S) 86%. This is not reflected in the other studies. VersionOne states pair programming is used in 24% of teams. [42], [45]

The thesis work from Vuorinen mentioned that testing was experienced as difficult task and not working very well with Scrum [42]. The questionnaire in this thesis had no similar findings. The Vuorinen thesis is from 2010; perhaps testing tools and habits have evolved to support Scrum practices with continuous delivering and testing. From the article research it can be seen that unit driven testing and TDD are commonly used with Scrum [11], [23]. The VersionOne survey shows unit tests used by 63% and TDD33% [45].

4.6 Tools comparison with questionnaire and articles

The highest tool usages measured by the VersionOne survey were task board 82%, bug tracker 80%, spreadsheet 74%, and the agile project management tool 71%; following these were the unit test tool 66% and the continuous integration tool 57%. Compared to tool usage findings from the questionnaire, it is interesting that no one mentioned spreadsheets and only one mentioned task board, which were the top 2 tools in the VersionOne survey. Other tools such agile project management tools were used by all the respondents to the questionnaire. The questionnaire showed that the most common agile management tool was JIRA. VersionOne survey states MS Excel is the top tool with 60%, and JIRA is second with 51%. JIRA findings are very much aligned between VersionOne and the questionnaire, since 50% of questionnaire respondents said to use JIRA. [45]

As regards the tools survey article findings, the most common tools were physical wall and paper 26% and spread sheets 23% [12]. JIRA was only used by 2%. This research is from 2011, which could explain the difference in JIRA usage. It could be that in the past more simple tools were used, and now more comprehensive tools exist like JIRA which can give status, visual reports, and access to anyone to see how the Sprint is proceeding. The tools survey states that best aspects of tool usage were ease of use and customizability, which is true with wall & paper and spreadsheets [12]. Issues were seen with lack of integration of other systems, custom reports, absence of a visual task board, and the inability to shift from high level reporting more detailed one. These issues are solved by using some agile project management tool such as IceScrum or JIRA, which can explain the high increase of the use of JIRA and other agile project management programs.

4.7 Perceptions of using Scrum by articles and questionnaire

Positive and negative aspects of Scrum in the background material as compared to questionnaire findings seem to be very much in alignment. In both the thesis questionnaire and the background material, the same issues were found to be positive sides of Scrum,

such as customer focus, transparency, flexibility, improved estimation, better communication, Planning Poker for estimations, continuous learning and team effort and improved team spirit; adding new features during the project but freezing the Sprint for work; commitment to Sprint goal, deliverable product in each Sprint, more confidence on project success; inspect and adjust in each Sprint, improved performance and more satisfied developers, better quality code and less error found.

Scrum challenges, issues that do not work well or cause problems mentioned both in the questionnaire responses and the research articles were, for example, prolonged meetings often taking valuable coding time, too much detail discussed in the Daily Scrum, and doing things without clear purpose such as having a Retrospective meeting but not following up on issues mentioned there—such problems can demotivate people. Further problems arose when a lack of Scrum knowledge by customers, management or even in the Development Team could cause issues with working with Scrum. Also nonfunctional requirements were mentioned to be hard to handle with Scrum. Over-planning tasks for Sprint can cause demotivation. Fixed priced contracts do not work well and are a legacy from the Waterfall world. While sometimes too many team meetings were scheduled for the 2 weeks Sprint, some teams had the opposite problem of not having enough meetings. Missing management support was found an issue too. All these difficulties were found in questionnaire results as well as studied articles.

In the VersionOne survey, some issues arose that were not mentioned as such in the questionnaire findings, like the ability to change organizational culture and general organizational resistance, the pre-existing rigid/Waterfall framework, and not enough resources for necessary agile experience. This is anyway reflected in the findings from the questionnaire through other answers lamenting lack of management support and deliveries coming from waterfall teams, lack of Scrum understanding among management and customers. All these problems often have the same root cause: that the company culture does not support using Scrum. Other topics that did not arise in the questionnaire findings but were mentioned in couple of the studies were geographical distance and cultural issues, both of which can cause problems when using Scrum. Also it was mentioned in background articles that it requires a lot of discipline to practice Scrum correctly. Some of the papers had studied teams with poor team spirit and limited understanding of Scrum. These issues were not seen either in questionnaire answers.

To sum up, problems arise under the following conditions: the organizational culture does not support Scrum, there is not enough knowledge as to how Scrum works and the Scrum practices are not respected by other teams or stakeholders, team work is being interrupted, and teams are managed from top down and not given freedom to work in a self-organized fashion. There are also issues rising from the fact that some of the Scrum practices are not employed well, such as the role of Scrum Master or some of the Scrum events; this could be also due to limited Scrum knowledge. Another reason is that Scrum does not specify how the work should be done, for example in the areas of architectural design or

testing or organizing the team. It is up to the team to decide such things, and thus, it can be challenging for each team to find what works for it. . Since Scrum is very people-centered, it very much depends on what kind of people one has working on the Scrum Team. Further dilemmas can arise when the customer is not involved in the process as Scrum would require, and hence the customer might not approve of how work is being done or does not understand the usage of Scrum.

Positive perceptions mostly arise from the three pillars of Scrum: *1) transparency, common language and definition of done 2) inspection, inspect Scrum artifacts towards the Sprint Goal and 3) adaptation, if in inspection it is noticed one or more aspects are not within limits, the process needs to be adapted to fix the course towards accepted area.* Root causes of positive perception are linked to these, like more interaction and communication and constant inspection and adaptation, flexibility, and learning. No work is done in vain, and the team focuses on the relevant, hence comes team satisfaction. All stakeholders can see the results quickly, and the transparent process gives good visibility, more confidence of project success, and team motivation.

5. EVALUATION

5.1 Research questions revisited

- 1) What Scrum practices are used in the field and how they are employed?

This thesis research succeeded in finding out what Scrum practices are used and also in getting more detailed data on how they are employed by asking respondents for their perceptions. It is important to know the perceptions of the Scrum Team members to shed more light on why something is seen to work or why something does not work. This gives valuable information to correct practices and focus on making some practices more desirable as well as easier to use. Quantitative comparison was also possible with other studies.

After closing the survey and going through the responses, one thought arose concerning the Product Owner. More information could have been asked about the Product Owner role. Information about this role was mostly retrieved via other questions, for example, those relating to the Burndown Chart and Product Backlog. The Scrum Master role was researched much more in this survey than that of the Product Owner.

- 2) What engineering practices are used along with Scrum?

From the research, it became clear as to what the most common engineering practices used in the companies of respondents are. The engineering practices that enable the team to work with iterations and deliver working software each Sprint were weekly builds, continuous integration and unit testing as mentioned in Chapter 4. The limitations to these questions were mentioned previously: the engineering practices were specified beforehand by article research findings and did not give change to specify if some other were used. It also gave mostly similar results as other research on the field but also some deviant data, peer review was found very popular in this research. So this thesis has brought forth some new findings in the area of Scrum in relation to used engineering practices.

- 3) What are the positive and negative aspects of using Scrum?

This question is linked to the first one, providing more depth to the questions about what Scrum practices are used. Respondents were given open questions to answer in own words to questions related for example Scrum events, Definitions of Done, and role of the Scrum Master. It was a successful approach, since it gave more information about how are certain practices perceived and if such perceptions affect the usage of those practices. For example, making estimations was thought to be hard sometimes, and correspondingly, some of the teams reported a lot of estimation errors. With more in-depth

information, more relationships between perception and implementation can be established.

4) What are the outcomes of using Scrum?

The outcomes of using Scrum were in line with the findings with other studies in the field. It can be concluded that those findings are the most widespread benefits of Scrum. The outcomes questions were multiple-choice, making possible easy data comparison with other studies such as the VersionOne survey. The question format was a good choice for comparison to other studies, but should have provided an extra slot for respondents to add other information/outcomes should they have some in mind.. All available options in the thesis questionnaire were specified by the questionnaire author.

5.2 Uncertainty factors

The main focus of this thesis is to research actual Scrum practices used and the engineering practices that are used alongside them. Background research materials did not pose the same questions as this thesis, so some points not mentioned in the articles might change the employment percentage of some practices. Articles were read through in order to collect all the mentioned practices and note the good and bad aspects of Scrum, but since most authors were not focused solely on practices, they may have overlooked and failed to mention some practices used. Based on the papers it cannot be concluded 100% that some specific practices were not used; one can only safely state that they were not mentioned in the research.

The questionnaire did not ask how long each specific team had been working with Scrum, but only the Scrum experience of the respondent. How long the team has been using Scrum affects the maturity of the team and also probably perceptions of good and bad aspects of Scrum. One team mentioned it was very new to Scrum but other teams' experience is unknown.

There was a multiple-option question to choose used engineering practices. The questionnaire could have had an extra question for some other used engineering practices. With this questionnaire, it was not possible for respondents to mention if some additional engineering practices were used beyond the ones mentioned in the multiple choices. The same applies to Scrum outcomes question. Thirdly there was no – “*Do you have a dedicated Product Owner?*”- question. The answers of the respondents in 10 papers mentioned that a PO is used, however for 4 teams, it was unclear.

Another point that can be stated as uncertainty factor is that the 10th annual state of agile survey is done by VersionOne, which is an agile project management tool vendor [41]. It can be argued if the survey is objective enough, since it has not been clearly described how the survey has been done and it is done by one vendor in the market. Vendor which

could tweak the results to show up better in the listings. The data used in this thesis from that survey, does not anyhow include any data related to the actual agile project management tool called VersionOne. Survey has collected answers from thousands of respondents so it is seen as a good enough source to compare.

5.3 Limitations of the research

This thesis researched 14 different teams in 12 companies. The number of answers was not very high, and also there were no other responses from the same teams to give assurance of the validity of the answers. The inquiry was limited by its approach to finding respondents, which was via industry contacts since there was no direct contact to company and management which could have enabled more responses from the same teams. Having more answers could have given better accuracy and given the questionnaire and subsequently the thesis a bigger opportunity to find significant differences [38].

The time scale of the thesis work was limited, so it was not possible to wait for a long time for more answers for the questionnaire. The online survey enabled multiple answers from several different countries, but it would have been possible to get more if there would have been a longer time frame to search for respondents. So, the time factor was a limitation for this research.

It was not easy to find papers focusing on the usage of Scrum practices, which limited the research article amount. Background material was only searched through IEEE Xplore database. Using some other databases could have given some alternative studies to compare to those found on IEEE Xplore. Nevertheless, it was clear that there are not many scientific studies of the usage of Scrum practices.

5.4 Future work

In the future, it would be interesting to pursue similar research for one of the companies that would involve multiple respondents from several teams, but also several respondents from same team; one could research and analyze those responses. Having more people answering from the same team gives more assurance of the validity of the answers, and answers from same company can shed light on some cultural issues that could affect the employment of Scrum practices. Based on such answers, one could give recommendations for improving the working methods with Scrum. In this thesis the improvement suggestions are not taking into consideration the whole picture from one team or company, just the actual Scrum practice as compared to the textbook Scrum, to see if there could be some benefit to using a stricter Scrum approach.

Based on this thesis, it can be seen there are some practices that are not so widely employed; one wonders why that is so. While the questionnaire did ask why some specific

practices are not used, it could also be interesting to know, if there are some known reasons to leave out some Scrum practices. Ways to improve the employment of these practices could be studied as well. Such practices are, for example, measuring velocity, using the burndown chart, including the customer in Planning and Review meetings, and ROI (return on investment) based prioritization of backlog (this last practice was addressed in question in the evaluation version of the questionnaire, but since it was not understood by most trial version respondents, it was assumed that it must be something that is not promoted/used, and therefore it was left out in final version of questionnaire).

A third possibility for future work is that the same questionnaire could be expanded, and the respondents could be given more time to answer it, ensuring a bigger sample for the questionnaire. It is interesting to continue to research what practices are being used in many companies, why some are not being used at all, and why some practices are not employed as much as others.

6. CONCLUSIONS

This thesis shows that there are Scrum practices that have been employed very well in most companies; background material research supports this observation. Problems often arise from the way practices have been used; for example, the Scrum Master does not always nurture the Scrum practices and guarantee his team has best possible working environment and support to achieve Sprint goals. Burndown Charts are used, but they track hours and days, instead of measureable data of work having been done, as compared to, for example, stories, giving an idea of achieved functionality. Retrospective meetings are held, but no actions are taken to improve the working processes. There is a Product Owner, but he is managing the team from the top down deciding what tasks the team should do in Sprint. There are Review meetings, but customers do not attend.

Questionnaire answers also show in many of the companies, some practices were not employed at all, such as measuring velocity or using a Burndown Chart, or including the customer in the process through attendance of Planning and Review meetings. In this thesis, reasons why such practices have not been effectively employed has not been studied; this is fertile ground for future work.

According to the questionnaire, the top 6 employed practices are the Daily Scrum, Product Backlog, Sprint Backlog, fixed Sprint, Planning meeting and Scrum Master; over 90% of teams surveyed employ these. Research articles' top employed practices are the fixed Sprint, Daily Scrum, dedicated Product Owner, and Scrum Master; after these come backlogs and the rest of the Scrum events. The VersionOne survey finds the most employed practices are the Daily Scrum, prioritized backlog, short Sprints, Retrospectives and Planning meetings. Ultimately, this thesis' questionnaire, research findings, and VersionOne survey are very much in line with each other. Percentages vary a bit, but that is commensurate with the different approaches to collecting material and data.

Engineering practices used on the side with Scrum according to questionnaire findings are peer review, story points, continuous integration, and weekly builds, test driven development and Planning Poker. Peer review is the most common practice, and surprisingly it is not found in any of the background material. Other practices are also mentioned in background material like Kanban and pair programming, which were not asked about in the questionnaire and hence it is not certain if they are used in any of the companies answering the questionnaire.

There are many positive sides using Scrum but still there are also challenges that teams need to face. The findings from questionnaire correlate very well with the findings from article research on earlier studies. A few of the positive aspects mentioned both in back-

ground articles and the questionnaire are transparency, better team work and communication, and focus. Negative aspects are, among others, management support, architectural design, estimation, and interruptions. These aspects suffer from the fact that often organization culture does not support Scrum usage and Scrum knowledge is limited. Version one stated the top one reason for having issues with Scrum was the inability to change organizational culture.

Based on the article research and questionnaire findings, it is clear that there are teams that use Scrum almost by the book but there are some which are altering the practices to suit their needs; they can still find improvements in their practices. There are also multiple studies that show that Scrum has been altered, and companies are making their own versions of it. Maybe textbook Scrum is not for everyone; it is said that Scrum Teams should consist of highly skilled motivated members who need to be very disciplined to hold all the meetings and deliver Sprint goals [31], [37]. Based on the research, it can be stated that in using only parts of Scrum, improvements can still be introduced. For example Daily Scrum brings knowledge sharing, more communication, team work and a forum for bringing forward any impediments. Everyone is up-to-date about project status. Another highly employed part of Scrum is a prioritized and changing backlog that gives flexibility and focuses on customer needs.

This thesis has collected data from the field and pointed out Scrum practices that are seen to bring value to current corporate software programming work, and also point out those practices that are less employed. Based on the research, teams can look at their own practices, and see if they would benefit from more close adaptation of textbook Scrum. Future work remains to research less-employed practices and the reasons behind. This thesis work has also pointed out the good outcomes of Scrum as well as its challenges. Challenges are usually such as arise from the bigger picture of organizational culture and overall acceptance of using Scrum. Here management has work to do to acknowledge its responsibility and look at the gains achievable by working in a more agile way as a company supported from top to down. Scrum is a people-centred methodology and the success of it is very much depending on both the people using it and the people surrounding the Scrum Team.

REFERENCES

- [1] Advantages and disadvantages of questionnaire [WWW] URL: http://libweb.surrey.ac.uk/library/skills/Introduction%20to%20Research%20and%20Managing%20Information%20Leicester/page_51.htm , referred 13.2.2016
- [2] AGILEFAQ What is a story point? [WWW] URL: <https://agilefaq.wordpress.com/2007/11/13/what-is-a-story-point/>, referred 16.3.2016
- [3] Ani Liza Asnawi, Andrew M. Gravell and Gary B. Wills, Emergence of Agile Methods: Perceptions from Sw Practitioners in Malaysia, Agile India 2012 Conference, 2012, pp. 30-39.
- [4] Apache Subversion SVN [WWW], URL: <https://subversion.apache.org/>, referred 24.4.2016
- [5] Beck Kent, Beedle, Mike, van Bennekum Arie, Cockburn Alistair, Cunningham Ward, Fowler Martin, Grenning James, Highsmith Jim, Hunt Andrew, Jeffries Ron, Kern Jon, Marick Brian, Martin Robert C., Mellor Steve, Schwaber Ken, Sutherland Jeff, Thomas Dave, Manifesto for Agile Sw Development, 2001, URL: <http://www.agilemanifesto.org/>, referred 19.2.2016
- [6] Breno Lisi Romano, Alan Delgado da Silva, Project management using the Scrum agile method: A case study within a small enterprise, 2015 12th International Conference on Information Technology - New Generations, 2015, pp. 774-776
- [7] Burn-Down Chart: An Effective Planning and Tracking Tool [WWW], URL: <https://www.Scrumalliance.org/community/articles/2013/august/burn-down-chart-%E2%80%93-an-effective-planning-and-tracki>, referred 29.4.2016
- [8] Daniel Pauly, Bjoern Michalik, Dirk Basten, Do Daily Scrums Have to Take Place Each Day? A Case Study of Customized Scrum Principles at an E-Commerce Company, 2015 48th Hawaii International Conference on System Sciences, 2015, pp. 5074-5083
- [9] David Bustard, George Wilkie, Des Greer, The Maturation of Agile Sw Development Principles and Practice: Observations on Successive Industrial Studies in 2010 and 2012, 20th Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems (ECBS), 2013, pp. 139-146
- [10] Diego Lo Giudice with Holger Kisker, Ph.D., Nasry Angel, How Can You Scale Your Agile Adoption?, Forrester, 2014, 17 p.

- [11] Franz Zieris, Stephan Salinger, Doing Scrum Rather Than Being Agile: A Case Study on Actual Nearshoring Practice, 2013 IEEE 8th International Conference on Global Sw Engineering, 2013, pp. 144-153
- [12] Gayane Azizyan, Miganoush Katrin Magarian, Mira Kajko-Mattson, Survey of Agile Tool Usage and Needs, 2011 Agile Conference, 2011, pp. 29-38
- [13] George A. Miller, The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information, First published in Psychological Review, 63 pp.81-97, [WWW], URL: <http://homepage.psy.utexas.edu/homepage/class/Psy355/Gilden/MagicNumberSeven.pdf>, referred 16.2.2016
- [14] GIT – fast version control [WWW], URL: <https://git-scm.com/>, referred 24.4.2016
- [15] Haikala Ilkka, Mikkonen Tommi, Ohjelmistotuotannon käytännöt, Talentum Media Oy, 2011, 242 p.
- [16] Highsmith Jim, History: The Agile Manifesto [WWW], 2011, URL: <http://www.agilemanifesto.org/history.html>, referred 19.2.2016
- [17] Hirotaka Takeuchi, Ikujiro Nonaka, The New New Product Development Game, Harvard Business Review January 1986 Issue, 1986, URL: <https://hbr.org/1986/01/the-new-new-product-development-game>, referred 16.2.2016
- [18] History of Agile [WWW], URL: <http://training.leadingagile.com/pmi-agile-certified-practitioner/>, referred 26.4.2016
- [19] HP Application Lifecycle Management [WWW], URL: http://www8.hp.com/us/en/sw-solutions/application-lifecycle-management.html?jumpid=va_2igcjke9mp, referred 8.4.2014
- [20] IceScrum Open Source Scrum & Agile project management tool [WWW], URL: <https://www.iceScrum.com/>, referred 24.4.2016
- [21] JAMA a modern solution for product development [WWW], URL: <http://www.jamasw.com/>, referred 26.4.2016
- [22] JIRA [WWW], URL: <https://www.atlassian.com/sw/jira>, referred 8.4.2016
- [23] Laurie Williams, Gabe Brown, Adam Meltzer, Nachiappan Nagappan, Scrum + Engineering Practices: Experiences of Three Microsoft Teams, 2011 International Symposium on Empirical Sw Engineering and Measurement, 2011, pp. 463-471
- [24] Lucas Gren, Richard Torkar, Robert Feldt, Work Motivational Challenges Regarding the Interface Between Agile Teams and a Non-Agile Surrounding Organization: A case study, 2014 Agile Conference, 2014, pp. 11-15

- [25] Mens, Tom, Demeyer, Serge, *Sw Evolution*, Springer-Verlag Berlin Heidelberg, 2008, 347 p.
- [26] Paul L. Bannerman, Emam Hossain, Ross Jeffery, *Scrum Practice Mitigation of Global Sw Development Coordination Challenges: A Distinctive Advantage?*, 2012 45th Hawaii International Conference on System Sciences, 2012, pp. 5309-5318
- [27] Ricardo Y. Takahira, Lilian R. Laraia, Frederico A. Dias, Abraham S. Yu, Paulo T. S. Nascimento, Alceu S. Camargo Jr, *Scrum and Embedded Sw Development for the Automotive Industry*, 2014 Proceedings of PICMET '14: Infrastructure and Service Integration., 2014, pp. 2664-2672
- [28] R.K. Chandana Ranasinghe and Indika Perera, *Effectiveness of Scrum for Offshore Sw Development in Sri Lanka*, Moratuwa Engineering Research Conference (MERCCon), 2015, pp. 306-311
- [29] Rüssel Felix, *ScrumButt bzw. Nokia-Test als Excel-Tabelle*, Armerkater.de [WWW], 2009 URL: <http://www.armerkater.de/2009/01/Scrumbutt-bzw-nokia-test-als-tabelle/>, referred 11.2.2016
- [30] Sarah Beecham, John Noll, Ita Richardson, *Using Agile practices to solve Global Sw Development problems – A Case Study*, 2014 IEEE International Conference on Global Sw Engineering Workshops, 2014, pp. 5-10
- [31] Schwaber Ken, Beedle Mike, *Agile Sw Development with Scrum: Pearson International Edition*, Prentice Hall, 2002, 158 p.
- [32] *Scrum Framework*, Agileforall.com [WWW], URL: http://agileforall.com/wp-content/uploads/2013/05/Scrum_Framework.jpg, referred 9.3.2016
- [33] *Skype* [WWW], URL: <https://www.skype.com/de/>, referred 24.4.2016
- [34] *Slack iceScrum.com* [WWW], URL: <https://www.iceScrum.com/documentation/slack/>, referred 2016
- [35] *SurveyMonkey* [WWW], URL: <https://www.surveymonkey.com/home/>, referred 16.2.2016
- [36] Sutherland Jeff, Schwaber Ken, *The Scrum Guide™: The Definitive Guide to Scrum: The Rules of the Game* [WWW], 2013, URL: <http://www.Scrum-guides.org/docs/Scrumguide/v1/Scrum-Guide-US.pdf#zoom=100>, referred 19.2.2016
- [37] Sven Overhage, Sebastian Schlauderer, *Investigating the Long-Term Acceptance of Agile Methodologies: An Empirical Study of Developer Perceptions in Scrum Projects*, 2012 45th Hawaii International Conference on System Sciences, 2012, pp. 5452-5461

- [38] The importance of n (sample size) in Statistics, Statistics for the Terrified [WWW], URL: <http://www.conceptstew.co.uk/pages/nsamplesize.html>, referred 28.4.2016
- [39] User Stories Mountain Goat SW [WWW], URL: <https://www.mountangoatsw.com/agile/user-stories>, referred 26.4.2016
- [40] Velocity - VersionOne, Measuring the Velocity of your Agile Scrum Team [WWW], URL: <https://www.VersionOne.com/agile-101/agile-project-management-customer-management-best-practices/agile-Scrum-velocity/>, referred 12.3.2016
- [41] VersionOne Enterprise Agile Platform [WWW], URL: <https://www.VersionOne.com/>, referred 28.4.2016
- [42] Vuorinen Jyri, Scrum-menetelmän käyttö Pirkanmaalaisissa ohjelmistoyrityksissä, Diplomityö, 2010, 45 p URL: <http://dspace.cc.tut.fi/dpub/handle/123456789/6888>, referred 12.3.2016
- [43] Waters, Kelly, How agile are you? allaboutagile.com [WWW], 2008, URL: <http://www.allaboutagile.com/how-agile-are-you-take-this-42-point-test/>, referred 11.2.2016
- [44] 6 Thinking Hats Retrospective [WWW], URL: http://retrospectivewiki.org/index.php?title=6_Thinking_Hats_Retrospective, referred 27.4.2016
- [45] 10th Annual State of Agile report , VersionOne Inc [WWW], URL: <http://stateofagile.VersionOne.com/>, referred 2.4.2016
- .

APPENDIX A: TEMPLATE FOR DATA COLLECTION FROM ARTICLES

Study
Year to publish
Bibliographic reference
Type of article
The aim of study
Type of study
Target country
Scrum experience
Findings
Good experiences
Challenges
Other specific findings & details

APPENDIX B: QUESTIONNAIRE

This questionnaire is a part of a research material for my Master Thesis: Scrum in practice - global review, done for SW Engineering faculty in Technical University of Tampere in Finland.

I would be very happy if you can help me to have more material for my thesis, and fill out this questionnaire. Please reply latest on 20th of March, so I can have time to analyze all material before doing conclusions for the research, sooner the better of course.

1. General info*

Company you work for? _____

Town and city? _____

Your job title and role in Scrum Team? _____

Size of Scrum Team? _____

Team is local or distributed? _____

Your Scrum experience in total? _____

2. About Scrum*

What is the general feeling towards Scrum in your team?

What kind of feedback you get about using Scrum from you customer?

What are the leanings in your team when you have been using Scrum?

What you think are positive sides of Scrum? _____

What does not work with Scrum? _____

What are the challenges with Scrum? _____

3. Sprint

What is the Sprint length? _____

Is Sprint fixed length? _____

4. What is your definition of done? _____

5. What kind of testing you use? Choose all relevant options

- Unit testing
- Feature testing
- SW testing is part of the code delivery
- New code is tested to be compiling with new baseline
- Integrated code smoke tested in SW baseline

Any other remarks related on testing? _____

6. What kind of requirements you use? Choose all relevant

- Big requirement documents which are ready when project starts
- Requirements that change and develop quite much during the project
- User Stories

Any other remarks on requirements? _____

7. Scrum Master

What does Scrum Master do in your team? _____

Do you feel he/she helps to achieve the Sprint goal? _____

8. Product Backlog, choose valid options

- Your Product Backlog is estimated (approx time to finish)
- All estimates are produced by the team
- Estimates done by Planning Poker
- Estimates not asked from the team
- A lot of estimation errors happen
- Estimates are pretty accurate

9. Velocity

Do you know what is your teams velocity? _____

How your measure teams velocity? _____

10. Burndown Chart - choose all valid points

- You use Burndown Chart
- You do not use Burndown Chart
- Burndown Chart is updated by team members
- Burndown Chart tracks hours and days
- Burndown Chart tracks Tasks
- Burndown Chart tracks Stories Done

11. Team working ways - choose all points relevant

- Tasks are assigned by Scrum Master or someone else?
- Tasks are chosen individually by team members
- Team can work without any interruption during the Sprint
- Team gets interrupted many times during the Sprint
- Team gets interrupted few times during the Sprint

12. Sprint Planning meeting

Do you have Sprint Planning meeting? _____

How long it is and who attends? _____

Who decides what will be taken into next Sprint? _____

Is customer present?

Good and bad sides of the meeting _____

13. Sprint Do you have Review meeting? _____

How long it is and who attends? _____

What is presented in the meeting? _____

Is customer present? _____

Good and bad sides of the meeting Review meeting _____

14. Sprint Retrospective meeting

Do you have Retrospective meeting? _____

How long it is and who attends? _____

What are the topics you go through? _____

Good and bad sides of the meeting _____

15. Daily Scrum

Do you have Daily Scrum? _____

How often and who attends? _____

How long is the meeting? _____

What are the topics you go through? _____

Good and bad sides of the meeting _____

16. Which of the following practices you use? Choose all relevant points

- Story points
- Continuous integration
- Daily builds
- Weekly builds
- Unit-test-driven development
- Unit tests written before code
- Unit tests written after the code
- Quality gates
- Source control
- Code coverage
- Static analysis tool
- Peer review

XML documentation

17. Tools

Communication with the team

Code repository

Testing

Agile project management

Other tools

18. What points you think are valid for you and your team related to Scrum and using it. Choose all relevant points

Transparency is good in the project

Team work is getting better

Team velocity is known, so it is easy plan project durations

Code quality is good

Not too many errors

Lots of coding errors

Team members respect each other

Team is open to face and point out impediments

Impediments are being removed efficient

There are many impediments making work hard

Requirements are clear and not changing

Requirements change along the project

Project progress is being monitored with actual measurable data

Communication is open and constructive

Communication is not open and lot of issues dealing with team members

- Daily Scrum, Sprint Planning meeting, Review and Retrospective are the only meetings held
- Scrum Team is committed to the Sprint goal
- Scrum Team has free hands to meet the Sprint goal
- Scrum Team is being managed by some team leader or project manager, telling what is being done and when
- Team is improving its performance by using Scrum
- Continuous learning is happening with Scrum

19. Any other comments related to Scrum you like to add?
