View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by University of Oulu Repository - Jultika

Systematic mapping study of usability in post-implementation on agile software development

University of Oulu
Faculty of Information Technology and Electrical Engineering / M3S
Master's Thesis
Tommi Helala
04.06.2019

Abstract

Need for new software information systems is increasing year by year and information software systems have become present in everyday life of people. As the number of systems has increased so has the need of these systems to be usable and work properly. This thesis used systematic mapping study method to get overview of the current state of usability in agile software development. In the study, initially 269 papers were retrieved from SCOPUS and after exclusion of irrelevant papers 92 papers were selected to the study of which 75 papers got through inclusion criteria to the final stage of the study. In this thesis a look to current state of usability in agile software development is presented. Study suggested that usability usage in agile environment is still trying to find its place but there is research being done constantly to make it more prevalent in the field. From those agile software development projects, that had included the usage of some sort of usability method to development, too few included usability throughout the whole development cycle.

Keywords
Usability, agile software development, systematic mapping study

Supervisor
Raija Halonen

Foreword

My personal motivation for researching this topic sparked when I was working for a company that had its whole ERP system completely redone. Big plans were made with structured design and timeline. But as it can be in any development, not everything went as planned. Timeline stretched and when the system was initially implemented for its employees to use, it was changed massively from what it originally was meant to be. Workflows and order of tasks done had been modified greatly. Also, since the timeline had been stretched the training given to employees was done nearly year ago and to mostly different system as it was back then. This meant that employees had to re-learn the system completely and partly forget what they had been taught earlier. This in my mind highlighted the need for usability activities such as testing with users, requirements re-engineering with usability in mind and the actual workflows of the users.

Tommi Helala

Oulu, June 4, 2019

Abbreviations

SLR	Systematic Literature Review
SMS	Systematic Mapping Study
ASD	Agile Software Development
HCI	Human Computer Interaction
ISO	International Organization for Standardization
UCD	User-Centered Design
UCASD	User-Centered Agile Software Development
ERP	Enterprise Resource Planning

Contents

1. Introduction

The purpose of the study was to analyze how usability in agile software development is noted in literature. The motivation for the study was based on understanding that in the 21st century information systems have become something that is present in everyday life of people. Most people in developed countries use, act or work with some kind of software information system. These systems support the everyday actions we need to perform. Systems we use range from ATMs to systems that are necessary in our work environment such as enterprise systems that support sales, billing, resource planning.

These software information systems do not just appear from nowhere. Behind every software information system there is great amount of planning, designing, developing and implementing necessary to make these systems work as wanted. As the number of systems is increasing, there is increasing need to find a way to make these systems work in the end. One of the biggest pitfalls in software information system development is implementation and its execution. One of the risks identified in system implementation was established by Scott and Vessey who claimed that in majority of information system projects failures come from management issues surrounding the implementation (Scott & Vessey, 2002).

Focus from traditional software development methods has shifted towards Agile software development (ASD) and following that, the focus of usability research has shifted towards ASD. These methods were designed to aid in the ever-shifting requirements and constant changes that traditional development methods were not suitable to handle. (Ruparelia, 2010.) This shift followed the need to establish more suitable replacements that used iterative cycles and larger involvement of stakeholders (Nerur & Balijepally, 2007).

When research started in topic of usability in post-implementation it was evident that even though the information systems have been developed widely and there is much research available in the topic (Sohaib & Khan, 2010; Salvador, Nakasone, & Pow-Sang, 2014; Magues, Castro, & Acuna, 2017), but there is still some work needed to be done. For example, there are little to no research done when considering usability of the system post-implementation. The steps taken before implementation of the system are widely documented, and so is the concept of usability, but not the connection of these two in the post-implementation phase, in so called shakedown phase (Nah, Lau, & Kuang, 2001). For enterprise resource planning systems Markus and Tanis (2000) have identified four phase life cycle: Chartering, Project, Shakedown and lastly Onward and upward-phase. This study focused on the last two phases: Shakedown and Onward and upward.

In this study a description of issues involving the usage of usability methods in information system development is given. This study did not specifically focus on specific type of systems, as all the information systems can be considered to have similar steps, whether it is designed for enterprise or public use. Main research problem for this study was:

RQ: How is usability in agile software development noted in literature?

When trying to answer to this research problem, following sub-research questions where iterated for systematic mapping study (SMS) to help getting answers. First sub-research question:

RQ1: How many papers include usability within agile software development?

RQ2: At what stages of development were the usability methods applied?

RQ3: *To which extent is usability mentioned in the later stages of the development?*

SMS method was chosen for this study for its' benefits over more traditional systematic literature review (SLR). One main difference between the SMS and SLR is that SLR usually attempts to investigate primary studies to find out the research outcomes of those studies and work with the results. SMS is usually more focused on aiming to classify and providing overview of the field of selected research. (Kitchenham, Budgen, & Brereton, 2011.)

In this report, an overview of current state of usability in agile software development was presented. This was done by conducting SMS using Scopus database to retrieve articles and discussing the end results of SMS.

This thesis is structured as follows. First an overview and definitions of usability are given in Chapter 2. Then Chapter 3 gives a look to system development life cycle with view of waterfall model and then focuses on few popular agile methodologies. Chapter 4 presents prior research done within the area of usability in agile development. Chapter 5 starts with definition of SMS as it is presented in the current literature followed with presentation of its usage in this thesis work. Next Chapter 6 presents findings of SMS and discussion of findings. Lastly Chapter 7 includes conclusions of this thesis.

2. Usability in information system design

In this chapter, the concept of usability in information system design is introduced.

Usability as a term has a very broad spectrum of uses. First time the term usability was recorded can be dated back to 1800s made by Thomas De Quincey in 1842. De Quincey used it when trying to differentiate the term utility from usability. (Hertzum, 2010)

There are several different definitions of what usability means in Human computer interaction (HCI). In the paper by Hertzum (2010) the term usability is not seen as a definition, more as an independent view of an idea. Definition has been lost to translation in HCI as each professional has their own image what usability is. Hertzum describes, that in HCI the term usability has no definitive definition, it has become so ubiquitous that usability is used without the definition. Usability can be divided in partial images of usability that partially describe it. (Hertzum, 2010.) Six images of usability are pictured in Figure 1.

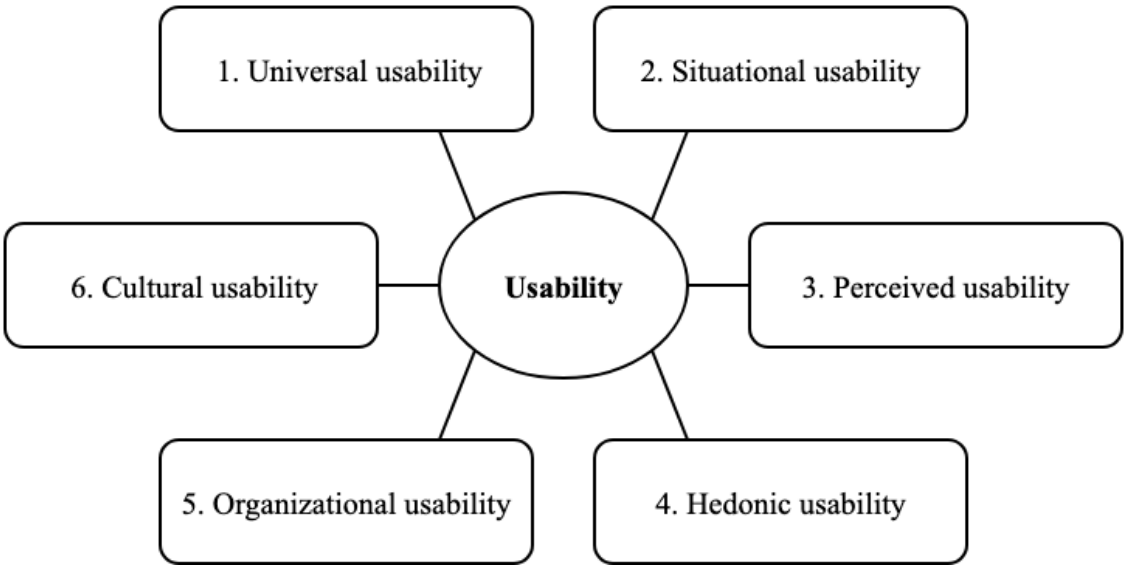

Figure 1. Six images of usability (Hertzum, 2010).

1. Universal usability – this means that universal usability is reached when maximum number of people can operate products in maximum number of different type of situations as is possible (Vanderheiden, 2000). Designers embrace the idea of global usability of system. This image is difficult to achieve since people are all different with different premises, but still the goal is to make everybody's experience successful. This type of usability is usually met in systems that are in global use for all demographic groups. In systems such as ATMs and healthcare. (Hertzum, 2010.) As three main challenges of this type are user diversity, knowledge gaps and technology variety (Shneiderman, 2000).

2. Situational usability – according to this image, usability is defined based on its equivalency to the quality-in-use of a system in certain context of use (Hertzum, 2010). This means that usability needs to be understood in relation to given people, tasks and context (Gould & Lewis, 1985). Situational usability comes closest to the ISO 9241 definition that is presented later in this chapter. Basically, situational usability means "…the quality of the use situation." (Hertzum, 2010).

3. Perceived usability – by this image the "…usability concerns the user's subjective experience of a system based on his or her interaction with it."(Hertzum, 2010). This image is the most user-centered approach towards usability. Perceived usability can also be identified as synonymous to perceived ease of use (Flavián, Guinalíu, & Gurrea, 2006). Therefore, its best used in situations where perceptions are primary and user performance is relevant factor (Hertzum, 2010). Paper published by Sonderegger and Sauer (2010) about influence of design aesthetics in usability settings suggested that the visual appearance increased the performance of system with reduced task times compared to less appealing system with identical usability features.

4. Hedonic usability – this image focuses on the feelings and emotions of their users instead of task-related qualities (Hassenzahl, Beu, & Burmester, 2001). This image means that the focus is on enjoyment of use rather than the ease of use, task accomplishment, and freedom of discomfort. As it is described in HCI the focus is that the tasks must be accomplishable and the ease of use of the system is priority, but hedonic usability turns this around. These priorities get over run by subjective experience of system use on focus to user pleasure and emotion. (Hertzum, 2010.)

5. Organizational usability – according to this image, usability is seen in organizational concept as a collaboration between groups of people. Focus is on corporate world as in the results and streamlining the work processes are prioritized over things like emotions and ease of use. Three elements are important to organizational usability: (1) There needs to be common ground among collaborators about goals, norms and their roles. (2) There needs to be awareness of the evolving state of collaboration in all work situations, and (3) there needs to be coordination of activities towards the performance of individual tasks. (Hertzum, 2010.)

6. Cultural usability – this usability considers the different cultural backgrounds of users. As in people with different cultural background consider usability differently as people from another culture. This comes relevant in situation such as designing global websites and global system user interfaces. Some colors in different cultures are associated to different things. Hertzum uses red as an example, since in United states it is associated with danger, but in China it is associated with happiness. (Hertzum, 2010.)

Another view of usability can be found from the standards of International Organization for Standardization (ISO). One of the definitions that derive from the ergonomics is the ISO 9241-11 standard (International Organization for Standardization, 2018). This definition is part of ISO 9241 series and its subsequently related to ergonomic standards. In ISO 9241-11 standards usability is defined as:

> *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."*

In ISO standards, there is also a more focused definition of usability in ISO/IEC 9126 (ISO, 1991) that focuses on supporting more of software engineering view of usability:

> *"A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users."*

From this ISO/IEC 9126 there has been a revised version ISO/IEC 9126-1 (ISO, 2001) that defines usability as follows:

> *"The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions."*

To this ISO/IEC 9126-1 definition the end, "…when used under specified conditions." was added so it becomes clearer that a product cannot have any intrinsic usability as it only has capability to be used in defined context. (Bevan, 2001.)

Usability can also be defined through system acceptability which was presented in the book written by Nielsen in 1994 (Nielsen, 1994). This view considers usability as a part of larger concept of system acceptability. Branch structure of system acceptability is presented in Figure 2.

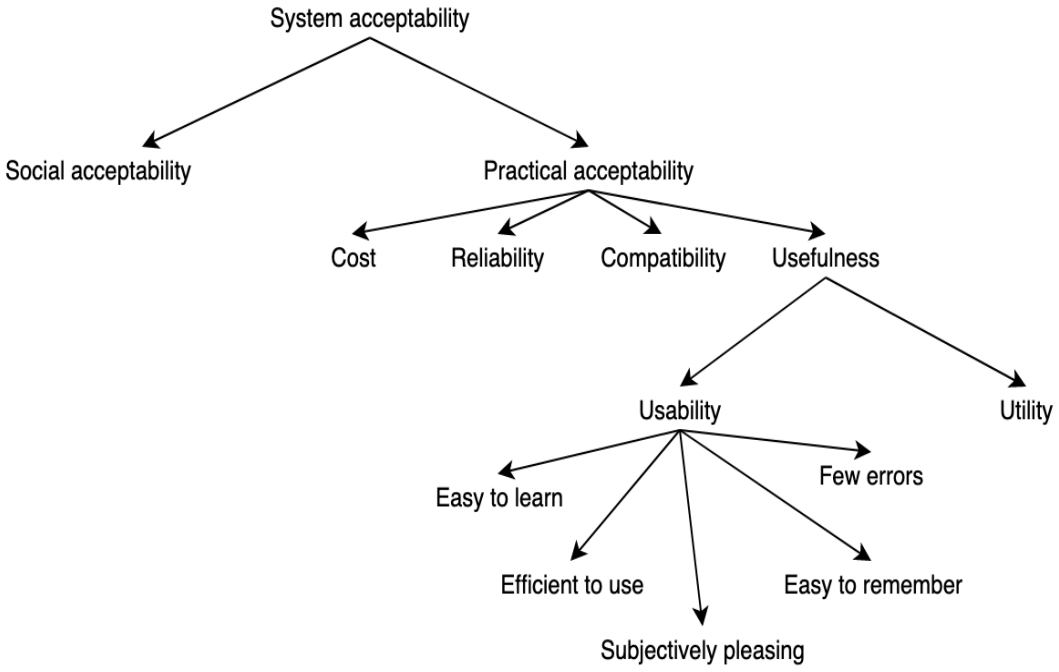

Figure 2. Model of system acceptability (Nielsen, 1994).

System acceptability describes the system good enough to satisfy all the needs of the users and stakeholders. Nielsen divides this further by splitting system acceptability to two branches: social acceptability and practical acceptability. Social acceptability focuses on evaluating user's thoughts such as is the system ethically and socially correct. Practical acceptability is focused more on the actual functionality of the system, not the consequences of its usage. Practical acceptability is covering things like cost, compatibility and reliability and usefulness of the system. Furthermore, one of the branches that reach out of practical acceptability is usefulness of the system. Usefulness is the main category between utility and usability. Utility is described to be as a "question of whether the functionality of the system in principle can do what is needed…" Here Nielsen describes the usability as a "…question of how well users can use that functionality." Some of the usability's attributes described are easy to learn, efficient to use, easy to remember, few errors and subjectively pleasing. (Nielsen, 1994.)

3. System development life cycle

In this chapter system development life cycle is introduced in two sub-chapters: waterfall model and agile methods. Agile methods are introduced as general and after that three popular methods are presented: Rapid application development (RAD), Scrum and Extreme programming

3.1 Waterfall model

In the early days of programming and development there were only two steps that took turns in development: write the code then fix the code. This would be like thinking about building to just hammer some wood together first, then taking a step back and looking what is wrong in it and if it identifies as a desired object. As it would be easily imagined, this development style had its problems. After few iterations of writing and fixing, the codes structure would fragment. It would not meet the requirements, because no requirements were taken to consideration and it would become long and expensive process to try test or modify the code. (Boehm, 1988.) Because, a better and more structured development process was needed.

One of the most fundamental system development models is the waterfall model. Waterfall model was first introduced in 1956 by Benington and was modified by Royce in 1970 (Benington, 1956; Royce, 1970). The original model was constructed from steps to be done one by one. These steps were originally in order: operational analysis, operational additional specification, design and coding specifications, development, testing, deployment and evaluation. Each step was done and evaluated at the end of the stage. It meant that the model was linear, and each step would be done only once. Royce saw this as a problem since the flow of process from the first step to next didn't have any chance of iteration. Lack of iteration could cause number of problems at the later stages of the development when the previous steps were not done effectively or correctly. Royce decided to introduce a feedback loop to the end of each stage to prevent this. This way at the end of each stage, there was a possibility to revisit the stage if needed. Royce also suggested that there needs to be something between requirements and analysis stage. Adding a phase between requirements and analysis would emphasize the minimizing effect of risks if the design phase would be done correctly. This model is presented in Figure 3. (Ruparelia, 2010.)

As there were several reasons that pushed the need for better development process there are issues in the waterfall model too that push the development processes to enhance and modify to suit current needs. Some of these issues were listed by Petersen, Wohlin and Baca (2009) as they conducted literature review. The main theme of these issues is the processes fitness for large scale projects, were it becomes rigid, expensive and high effort development process. It lacks opportunities for customers to interact and affect to the outcome of the system and problems that are not fixed early, pile up and introduce themselves in the later stages (Raccoon, 1997). The model can still have advantages too: early emphasis on the planning and predictability of the model are important when developing larges systems (Petersen et al., 2009).

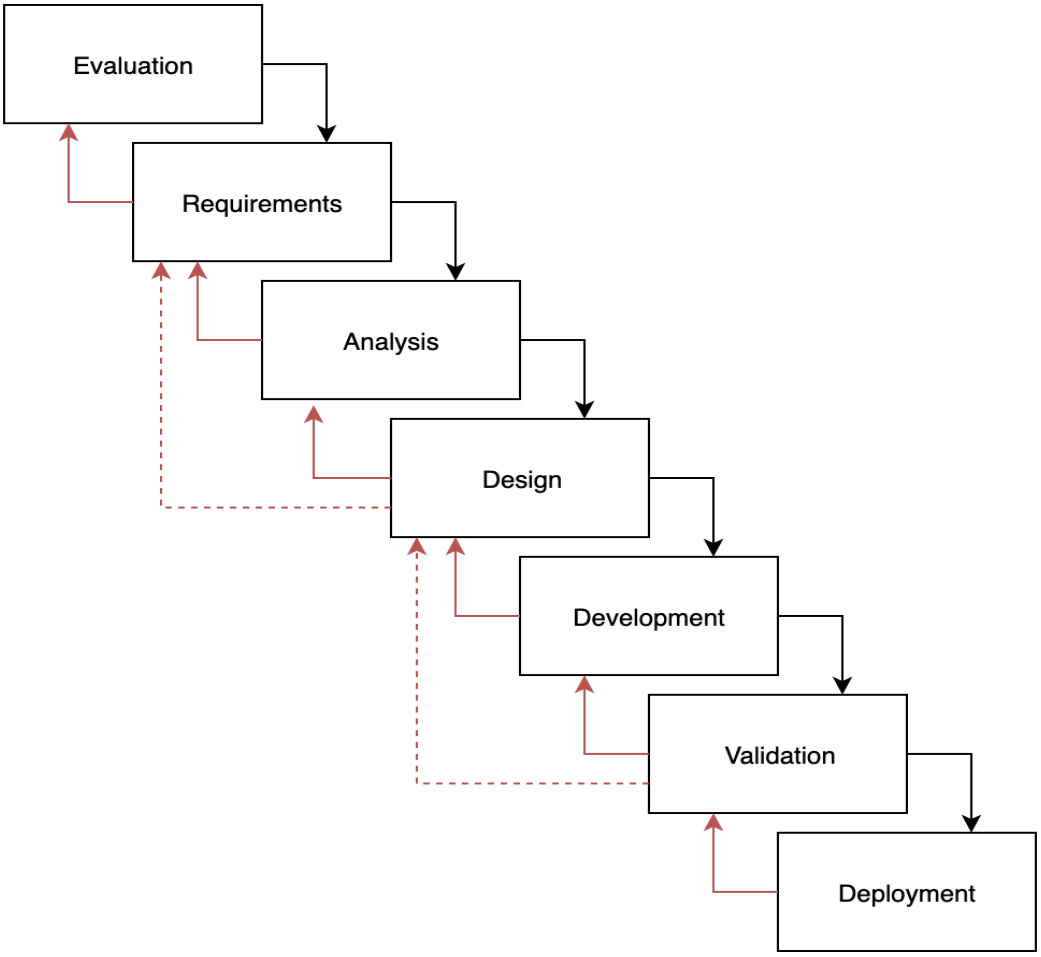

Figure 3. Royce's iterative waterfall model (Ruparelia, 2010).

The waterfall model gives this study the basic understanding of the processes that are done when developing information system. As the system development has evolved from the 1970's, there are plurality of different models that try to help developers when creating new systems (Ruparelia, 2010).

3.2 Agile methods

Principles described below are ones that at least agile methods, such as RAD, Scrum and XP, follow. These agile principles are described in Agile manifesto (Beck et al., 2001). These twelve principles are shortly described as follows:

1. Highest priority is to deliver early and continuous delivery of valuable software.
2. Welcome change
3. Deliver frequently.
4. Business people and developers work daily together.
5. Support the motivated individuals.
6. Promote face-to-face conversation between participants.
7. Key measurement is working software.
8. Promote sustainable development, keep constant pace.
9. Technical excellence and good design will speed up the process.
10. Maximize the work not done, Simplicity.
11. Promote teams to self-organize
12. Frequently gather together to consider how to become more effective and adjust accordingly.

All these methods follow the agile principles that focus on the constant change, small documentation and fast-phased releasing. (Ruparelia, 2010.)

RAD is Rapid Application Development that was developed by Martin in 1991. RAD is iterative methodology that uses prototyping as mechanism. It predates the agile manifesto and it can be considered as being and early type of agile development. RAD is presented in the figure 4. (Ruparelia, 2010.)

RAD is tailored towards reducing planning and encouraging more prototyping instead. This should enable more dynamic process and software should reach basic operational functionality faster. Also, reducing time between planning and writing the code limits the possibility of runaway projects. (Vuksanovic & Sudarevic, 2011.)

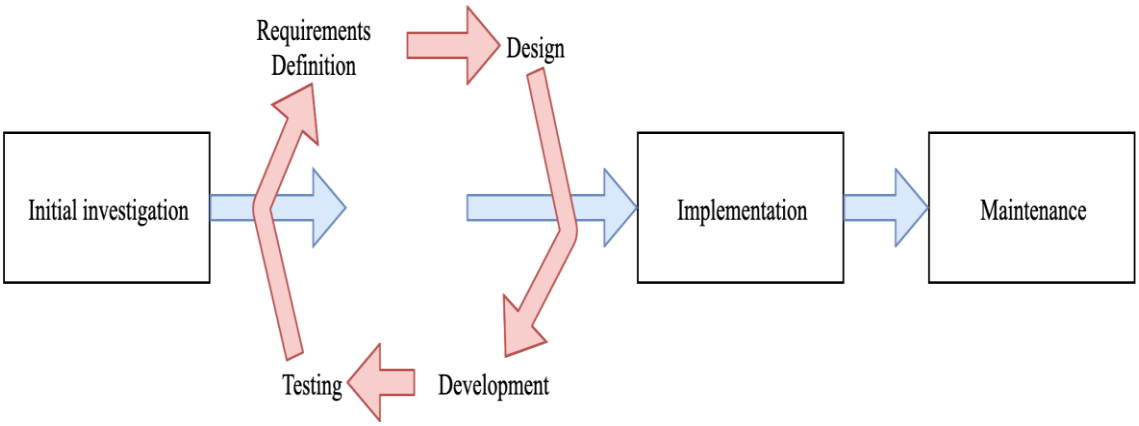

Figure 4. Rapid Application Development model (Ruparelia, 2010).

Scrum is a method that was developed to support development in small teams (Rising & Janoff, 2000). It is iterative and incremental methodology designed to deliver object-oriented software (Schwaber, 1997). Scrums strong point lies managing the volatility of software development and has empirical approach supporting flexibility, adaptability and productivity (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003). Main points of scrum are: First the product owner creates a priority list that describes the wishes and requirements for the software. From this list, the scrum team chooses the most prioritized pieces of software and focuses on developing those. The team has a fixed amount of time to deliver a chosen chunk of software. This time is usually between two to four weeks. This time is called sprint. With the team, there is ScrumMaster that makes sure that team is focused on the goals in each sprint. When the sprint is done the part of the software should be ready to be delivered to the customer. At the end of each sprint the team retrospectively reviews their achievement and methods in said sprint. After the first sprint is done the team chooses new chunk of software to their next sprint. This method is repeated until the software is complete. This is achieved when the priority list of the software is dealt with. (Sutherland & Schwaber, 2017.)

Extreme programming is an agile software development method introduced in the turn of the century (Beck, 1999; Beck & Gamma, 2000; Jeffries, Anderson, & Hendrickson, 2001). It is targeted towards small or medium sized teams in changing environment (Paulk, 2001). Method can be characterized as a fast-phased, agile method, that supports instant development and testing. Five main values of XP are communication, simplicity, feedback, courage and respect. These main values are followed in the practices of XP.

Next thirteen practices of XP are presented. Original practices had twelve steps, but refactoring was incorporated to incremental design. (Wells, 2017.) These practices are from the second edition of the XP practices:

1. First one is to sit together in the same space with the team to encourage communication.
2. Whole team needs to work together in a daily basis.
3. Workspace needs to be facilitated so that the face to face communication is encouraged and to keep team always informed with up-to-date information.
4. Work with effective way. Try not to overwork yourself or others and stay focused.
5. All production software is developed using pair programming.
6. Write user stories.
7. Weekly cycles need to be made. This is synonymous to an iteration.
8. Quarterly cycle is synonymous to a release.
9. Leave some extra task or stories that can be implemented later in your weekly cycle.
10. Software needs to build, and tests run within ten minutes.
11. Continuous integration to the code needs to be done.
12. Test-First programming needs to be followed. Write failing test-run failing test-developed code to pass the test-run test-repeat.
13. Practices of incremental design needs to be followed.

Extreme programming is somewhat similar to scrum and it has several similar aspects, but main difference comes from the Test-First programming practices. (Wells, 2017.)

4. Usability in agile software development

In this chapter a look at some previously done research regarding usability in agile development is provided.

Transformation from traditional linear style development to more iterative cycles of development and larger involvement of stakeholders has brought need to replace the currently used methods to once that suit better (Nerur & Balijepally, 2007). This need brought us Agile Software Development (ASD). Now that ASD has become mainstream there has been modifications done to its processes and some certain features are lacking from it. (Brhel, Meth, Maedche, & Werder, 2015.)

To further develop the foundations of ASD researchers have tried to combine ASD with User-Centered Design (UCD) and this combination is called User-Centered Agile Software Development (UCASD) (Brhel et al., 2015; Magues et al., 2017; Sohaib & Khan, 2010). There has been a multitude of different versions of this as some are trying to integrate them fully to each other and others are trying to take just the "best" parts of either UCD or ASD and mixing them. (Brhel et al., 2015) These mix methods are hard to get working since it is difficult to mix the fast and efficient agile methods with more thorough and labor-intensive tasks of UCD. Still both sides seem to acknowledge the failing of each method as their own. (Sohaib & Khan, 2010.) Table 1 shows the suggested combination of usability engineering and agile methods suggested by Sohaib & Khan (2010) from pages: V2-32 - V2-38.

Table 1. Combine approach of usability engineering and agile methods (Sohaib & Khan, 2010).

Agile methods Concepts	Usability Engineering	Suggested approach
Deliver working software frequently	Traditional software approach but iteration within phases	Iterative development throughout the project
Requires generalists	Requires specialists	Assemble a multidisciplinary team to ensure complete expertise
Customer focus	User focus	Collaboration between customers, users, product managers, Business analysts, developers, will maximize overall team efficiency for usable product
Test driven development and continuous integration	Contextual inquiry, field surveys, usability inspection methods for testing	Unit Testing + User Acceptance Testing + Usability testing throughout the process
Using onsite customer, functional requirements are encapsulated as user stories	Scenario based design for requirement analysis	Integrate user stories with scenario-based design

As it is shown in Table 1 there are five concepts that are suggested to be combined. First combination focused on improving delivery by combining agile-style frequent delivery with the iterative style of usability engineering. This translates to iterative development throughout the project. Second combination requires that multidisciplinary team is assembled to ensure that both styles have complete expertise available. Also, by combining customer focus and user focus greater efficiency was gained towards

developing usable product. Fourth approach combined test driven development and several usability testing methods and inquiries. Suggestion was usage of Unit testing, user acceptance testing and carrying the usability testing throughout the process. Last suggestion focused on integrating the user stories with scenario-based designs. These suggestions bring out the best of both styles of development and they can be compatible methods. (Sohaib & Khan, 2010.)

In 2017 Magues et al. published an SMS that focused on the current state of integration between agile processes and usability. Three databases were used and after usage of inclusion and exclusion criteria's 161 primary studies were selected that were published between 2002 and 2015. In the study they found out that integration approaches used in these papers could be divided in to four types. Processes that covered 47,83%, Practices that covered 19,25%, Team that covered 16,8% and finally Technology (4,34%). From this they concluded that there is still no clear agreement of the formalized method how the integration is done. Also, the increased number of publications since 2007 suggested that there is increased interest in this area. (Magues et al., 2017.)

Studies about usage of different usability methods in ASD have also been conducted in the last decade. Salvador, Nakasone and Pow-Snag (2014) conducted a systematic review about the use of usability techniques in ASD environment. 307 studies were identified through the search process but only 32 were selected for the final review. Study identified four most frequently used usability methods to be:

- Fast prototyping
- Individual inquiry
- Formal tests
- Heuristic evaluation.

Also, results of the study suggested that in 50% of the selected studies usability methods were used during the implementation phase. Implementation phase was here described to be "when either a prototype or the final version of the software is complete.". 40% also showed usage in design phase and only 22% during the requirements phase. (Salvador et al., 2014.) This is shown in Table 2.

Table 2. Usage of usability methods in ASD (Salvador et al., 2014).

Phase	% of studies
Requirements	21.88
Design	40.63
Implementation and Testing	50

Another development method that has been proposed was by Singh (2008) at Agile conference 2008. Singh proposed methodology called U-SCRUM to replace traditional Scrum method. Carrying idea of the U-SCRUM method was that more effort would be put towards usability of the product. This was done by placing second product owner called Usability product owner. Usability product owner would take over formulating the architectural and user experience vision. Basically, one product owner to focus on functionality and another on usability. (Singh, 2008.)

5. Systematic Mapping Study

Following chapters are structured as follows: First a definition of SMS is given according the literature and then a look at how the SMS has been executed in this study is given.

5.1 SMS Method in literature

SMS as a study method was widely used in medical research and later adopted in software engineering (Petersen, Feldt, Mujtaba, & Mattsson, 2008). In software engineering SMS is used to provide an overview of a topic area and to identify subtopics and gaps in current research for further study (Kitchenham et al., 2011). SMS can be seen following similar steps to SLR. Both base the information provided in the research on the literature, but steps that differ most are the classification of papers and data extraction and aggregation. (Petersen et al., 2008.) Table 3 provides differences between SMS and SLR methods.

Table 3. Differences between SMS and SLR (Kitchenham et al., 2011).

Elements of study	SMS	SLR
Goals	Classification and thematic analysis of literature on a software engineering topic.	Identifying best practice with respect to specific procedures, technologies, methods or tools by aggregating information from comparative studies.
Research question	Generic – related to research trends.	Specific – related to outcomes of empirical studies.
Search process	Defined by topic are.	Defined by research question which identifies the specific technologies being investigated.
Scope	Broad – all papers related to a topic are included but only classification data about these are collected.	Focused – only empirical papers related to a specific research are included and detailed information about individual research outcomes is extracted from each paper.
Search strategy requirements	Less stringent, authors can restrict themselves for more strict scope e.g. One or two digital libraries.	Extremely stringent – all relevant studies should be found.
Quality evaluation	Not essential.	Important to ensure that results are based on best quality evidence.
Results	A set of papers related to a topic area categorized in a variety of dimensions and counts of the number of papers in various categories	The outcomes of the primary studies are aggregated to answer the specific research question(s), possibly with qualifiers (e.g. results apply to novices only).

There are several benefits identified for conducting a SMS as a basis of follow-on research. Kitchenham et al. (2011) concluded that one of the significant benefits was the SMS's ability to set up a baseline for further research. It helps following researcher to have clear understanding of the current literature and therefore help them to set up their own research. Study also provides good baseline for comparing the follow-on results to

establish the legitimacy of the study. Also, further research can avoid the flaws and limitations of previous studies. (Kitchenham et al., 2011.)

On the other hand, problems could arise from doing follow-on research from SMS. Same paper (Kitchenham et al., 2011) identified some problems too. One of the biggest problems identified stems from the flaws and lack of quality in this initial study. If initial SMS was of poor quality, there could not be much to base the further research on. Also, if the flaws of the initial study are not well enough identified, they may carry on to further studies too. It also needs to be noted that if there is long time between the initial research and follow-on research the information gained from initial research may not be fully or at all usable and completely new search is needed. (Kitchenham et al., 2011.)

Petersen et al. (2008) identified five main steps for SMS. These steps are illustrated in the Figure 5 and in more detail in following chapters.

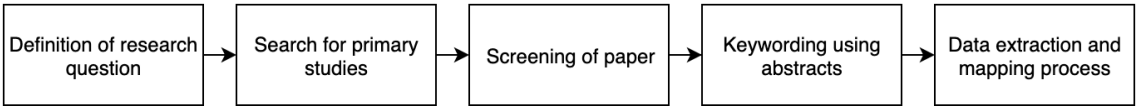

Figure 5. SMS process (Petersen et al., 2008).

Research Questions: First step is definition of research question. Research questions should reflect the goals of SMS, which are to provide overview of a research are, and identify quantity, type and results available within. (Petersen et al., 2008.)

Search for primary studies: Second step is conducting search for primary studies. Primary studies are identified by comprising a search string to selected databases or manually browsing publications and proceedings. Search string should be comprised from the research questions so it's relevance to study is clear. (Petersen et al., 2008.)

Screening of papers: Third step for the study is to screen the papers using selected inclusion and exclusion criteria to reveal the most relevant papers for the study. (Petersen et al., 2008.)

Keywording using abstracts: Fourth step is keywording of abstracts. This is done in first reading the abstracts and look for keywords and concepts. In this part you also identify the context of the research. After this is done, these selected keywords are combined to create bigger understanding of the nature and contribution of the research. If keyword identification cannot be done from the abstract, then reviewer can study other parts like introduction and conclusions from the papers. (Petersen et al., 2008.)

Data extraction and mapping process: Fifth step in SMS is data extraction and mapping of the studies. In this step the previously created classification scheme is used to map the studies and analyze the results of the study. (Petersen et al., 2008.)

To get an overview of how usability is used in agile development, a SMS is carried out. This was done over more traditional SLR since the research questions do not need the actual research outcomes of the studies to be answered. Main goal is to get an overview and classify the papers based on created classification. Also using this method, a broader amount of papers could be included (Kitchenham et al., 2011).

5.2 SMS in this study

The following chapters describe the steps that were taken in this study to conduct SMS.

5.2.1 Research questions

This study aimed to take look at current state of usability in agile development. Especially when the usability methods were used in the development, rather than what specific methods were used. To achieve this SMS was used as the research method to uncover, if the usability was used throughout the whole system development life cycle or just in specific stages. The goal of this study was defined in the following research questions:

RQ: How is usability in agile software development noted in literature?

As a main research problem this sentence described the general focus of this study. After choosing SMS as a research method for this study, following sub-questions described below were aimed to help find answer to main research question.

RQ1: How many papers include usability within agile software development?

First, an overview of the current usage of usability methods within the agile development practices was needed. From these papers start to gather information to proceed towards the goals of the study was gained.

RQ2: At what stages of development were the usability methods applied?

Second, after discovery of papers that had used usability in agile development information about used stages was found.

RQ3: To which extent is usability mentioned in the later stages of the development?

Third, from papers that clear identification of stages of usability methods used was done, an information if the usage was only limited to the requirements or the early phase of development was needed. This helped to determine that constant changes were put through usability methods to ensure usable and relevant product.

5.2.2 Search for the primary studies

Search for the papers was done by identifying search string to reflect the research questions. In this study the following search string was used:

TITLE-ABS-KEY (usability AND (“agile software development” OR scrum OR “rapid application development” OR rad OR “extreme programming” OR xp))

After the search string was comprised it was used in the selected database. Search was conducted on the SCOPUS database. SCOPUS database was selected for this study because it covers many other scientific publications such as ACM, IEEE and Elsevier. It also indexes other important software engineering sources like Empirical Software Engineering Journal, Springer Lecture Notes on Computer Science series and other publications and conference proceedings. Also, good search functions eased the process of going through the papers. Usage of the search string in the SCOPUS database gave 269 results. From this pool of papers, the inclusion and exclusion started for the study.

5.2.3 Screening of papers

Screening of primary studies was done using following inclusion and exclusion criteria:

Exclusion criteria:

- Removing of studies not written in English. (263left)
- Removing of other document types set by SCOPUS except conference papers and articles. (220 left)
- Removing studies that do not have abstract available (220 left)
- Removing studies not related to software usability and/or agile software development (165 left)
- Removing duplicated studies (164 left)
- Removing secondary studies, such as literature reviews, mapping studies and surveys. (147 left)
- Study has no full-text availability from either Google Scholar or from databases accessible to the students of University of Oulu. (129 left)
- Removing studies not relevant for the research questions. (92 left)

Inclusion criteria:

- From the study, a reviewer could identify at what stage the usability methods were used in the development. (75 left)

These exclusion and inclusion criteria where chosen to narrow the papers included in the study while trying to keep the all the relevant papers for the research question in the study. Article in this context means the papers categorized as an article by SCOPUS. Table 5 includes the usage numbers of exclusion and inclusion criteria. These criteria's also display some of the limitations of this study. Limitation include choosing only one language to search for the articles and by using just one reviewer to go through all the relevant papers to identify the stage of usability. This presented a limitation since there was a chance that some other reviewer could have identified the stage of usability.

5.2.4 Keywording using abstracts

Keywording using abstracts was conducted by identifying stages of development from previously done literature review. The selected categories were:

Chartering, Project, Shakedown and fourth category Onward and upward stage (Markus & Tanis, 2000). Terminology was taken from enterprise system experience cycle written by Markus and Tanis (2000). For this study phases were integrated to agile methodology to describe the development phases as follows:

- Chartering: This phase covers initial investigation and requirements definition. This is the projects starting line.
- Project: Project phase covers large area of development. It starts right after initial investigation and requirements definition. Some of the requirements are still iterated through this phase, but the main steps here are Testing, Development and Design. Also, this phase covers the initial implementation. This means the implementation that is first done for the system to be usable by users.

- Shakedown: This phase covers area after the initial implementation to being done with the primary goals of the development before moving to maintenance phase. Basically, from when the system is put to use until its usage is presumed normal operations.
- Onward and upward: Onward and upward phase describes the steps taken after the system has been implemented fully and is moved to maintenance stage.

These four stages can be put in agile software life cycle definition as presented in Figure 6.

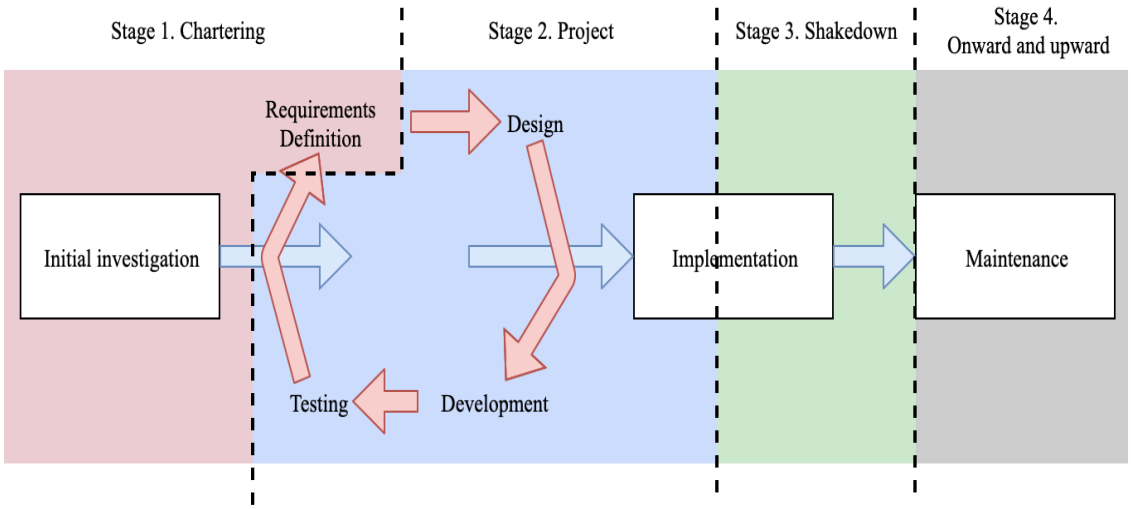

Figure 6. Rapid Application Development model including the stages of development.

Usage of the classification in this study was done with following rules:

- One development project could have used usability in one stage or in multiple stages.
- If the one development project had used usability methods in multiple stages, all stages that the method was used were marked with X in the spreadsheet.
- If the stage could not be identified by the researcher, it was marked with X to column named “Can’t be identified”.

5.2.5 Data extraction and mapping process

Data extraction was done with an excel sheet that included the studies that were left after using the exclusion criteria. From these articles the identification of inclusion criteria was done by reading the study until it was clear that either the stage or stages was identified, or it was not explicitly identifiable at what stage the usability methods were used. For the study items described in Table 4 were extracted from the data.

Table 4. Data items extracted from the papers.

Number of extracted items	Item name	Description
I1	Authors	Name of authors of the study
I2	Title	Name of the paper
I3	Year	Publication year of the study
I4	Source title	Name of the publication
I5	Cited by	Number of times cited according the SCOPUS
I6	Document type	Type of study
I7	What type	Snipped that describes the intent or the style of the study
I8	Chartering	Phase covers initial investigation and requirements definition
I9	Project	Phase covers most of the main development steps from requirements to initial implementation
I10	Shakedown	Phase covers after initial implementation to when "normal usage" of system is gained
I11	Onward and upward	Phase covers step from normal usage until the system is upgraded or replaced
I12	Can't be identified	Author cannot identify at what stage the usability methods were used, but they it is clear that they were used in the development
I13	Not relevant for RQ's	After further investigation, article was deemed not to be relevant subject area

All the other extracted items described in Table 4 are provided in the thesis with the exemption of I7. This items intention was to provide memory support for the researcher to identify the main points of the paper.

6. Findings

In this chapter, results of the conducted SMS are presented.

From the initial size of 269 papers, 92 were deemed relevant for the study and from those 92 papers stage could be identified from 75 papers. Table 5 shows the number of papers removed from the total number of papers and reason for papers removal.

Table 5. Number and reason of papers removal from the study.

Reason for removal of paper	Number of papers removed	Percentage of papers removed
Not in English	6	2%
Not a conference paper or article	43	16%
No abstract available	0	0%
Not related to software usability and/or agile software development	55	20%
Duplicate	1	< 1%
Not a primary study	17	6%
No full text-availability	18	7%
Not relevant for research questions	37	14%
Stage could not be identified	17	6%
Total number removed	194	72%

From Table 5 it can be seen that biggest reason for removal of paper from the study was that the paper was not related to software usability and/or agile software development. For this reason, 55 papers were removed from the study. Paper being not classified at SCOPUS as a conference paper or article was second biggest reason for removal with 43 papers removed. Third biggest reason was the relevancy towards research questions with 37 papers removed. As it can be seen from the Table 4 total number of papers removed from the study was 194 with that being 72% out of the total number of papers included initially.

Figure 7 shows the distribution of publication year that stage was identified from. No specific timeframe was chosen, instead the research questions and inclusion and exclusion criteria were used to narrow the results naturally. Papers that were used range from 2002 to 2018 with two gaps; one in 2003 and one gap in 2010.

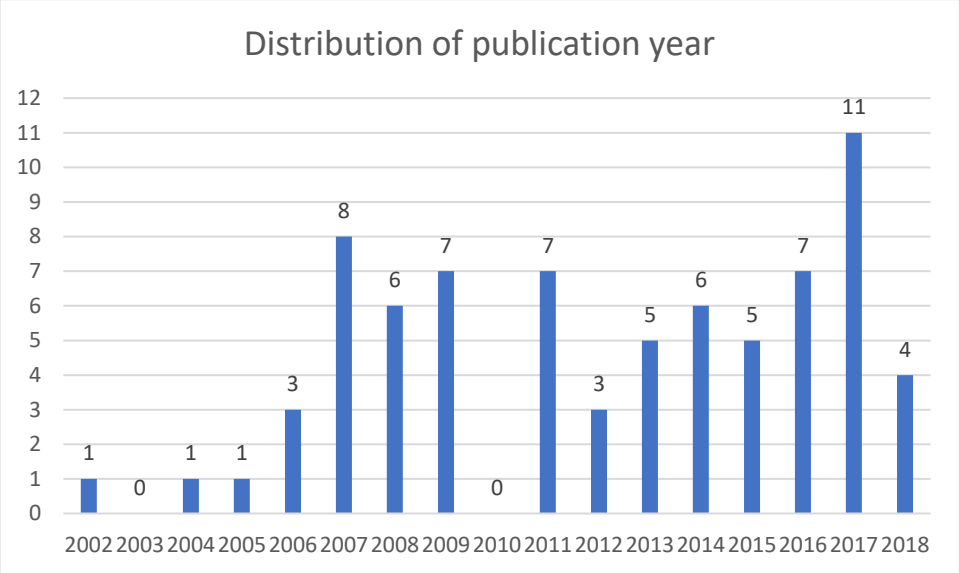

Figure 7. Distribution of publication year on papers that stage was identified

The main research problem in the study was to find out how and if the usability methods were used in the post-implementation.

RQ: How is usability in agile software development noted in literature?

To answer this, these 3 sub-questions were crafted to help out in the study. First sub-question was:

RQ1: How many papers include usability within agile software development?

Results from the study showed that from 2002 to 2018 there has been 92 papers written that acknowledge the usage of usability in agile development and this comes around 5,4 publications per year. This isn't that much compared to papers including agile development within same timeframe and including only articles and conference papers in SCOPUS and using the same search phrase excluding the word "usability". Conducting this search 7020 results were found which turns to around 412,9 publications per year.

Second sub-question crafted for SMS method was:

RQ2: At what stages of development were the usability methods applied?

Using the classification scheme the papers were categorized in four main classes: Chartering, Project, Shakedown, onward and upward. Figure 8 is showing the distribution of results in the study. Patch of 75 papers is 82% from the initial patch that cleared initial screening and exclusion criteria. This leaves the 18% of papers out based on the judgement of researcher. Figure 8 also shows that from the 75 papers, that the stage was identified from, 70 (93%) of those had some form of usability method used in the Chartering phase. Also 69 papers (92%) included project phase usability.

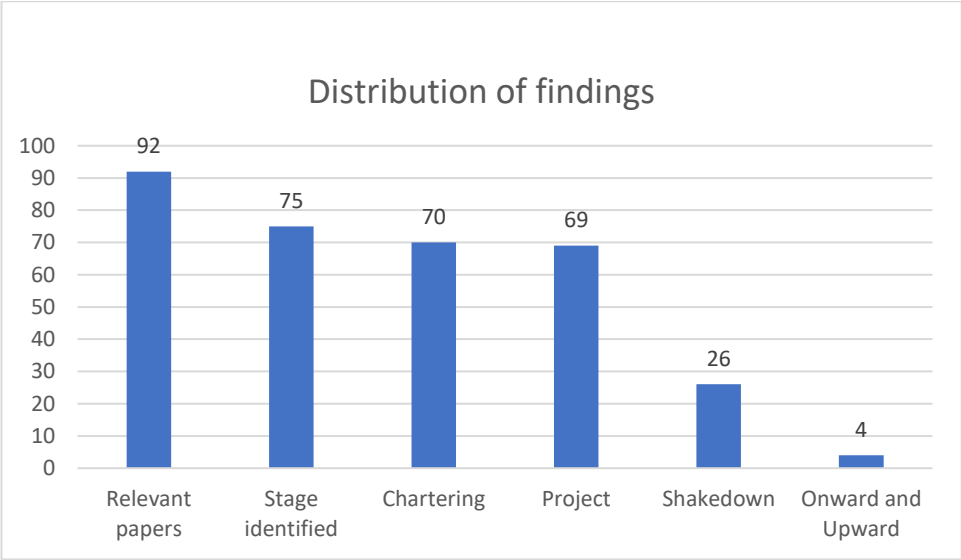

Figure 8. Distribution of findings.

Since the project phase is pretty comprehensive and covers most of the iterative development phase, not including the requirements gathering, it is also one of the phases that usability methods are most likely to be used. These two phases are also covered most in literature and remain as a focal point of usability method usage. In this study previously presented two stages are considered to be either on early or during the development up to the first implementation and later stages being after that initial implementation with focus on shakedown phase, since development should still be going nearly full force. Following the created classification results show that there was 26 (35%) papers that included usability in shakedown phase out of the papers that stage was identified from. In the last stage, onward and upward, only 4 (5%) papers included usability. These papers, as seen in the Appendix A, are all papers that also included usability in shakedown phase. Results of these two stages are covered more thoroughly in the next sub-question.

Third sub-question somewhat overlaps with previous questions, but it ties together the study. As it was previously mentioned the third sub-question was:

RQ3: To which extent is usability mentioned in the later stages of the development?

As it was mentioned in the results of second sub-question, shakedown phase usability was included in 26 papers that stage of usability was identified from. Distribution of per year is shown in Figure 8. As it can be seen from Figure 8, there are two spikes that covered this stage more than other years, but there is no clear trend to be identified from 2004 to 2018. For instance, in 2004 only one publication was deemed relevant and stage identified and from that article also shakedown stage usability usage was recognized but also at the same time in 2013 included five papers but none represented shakedown stage.

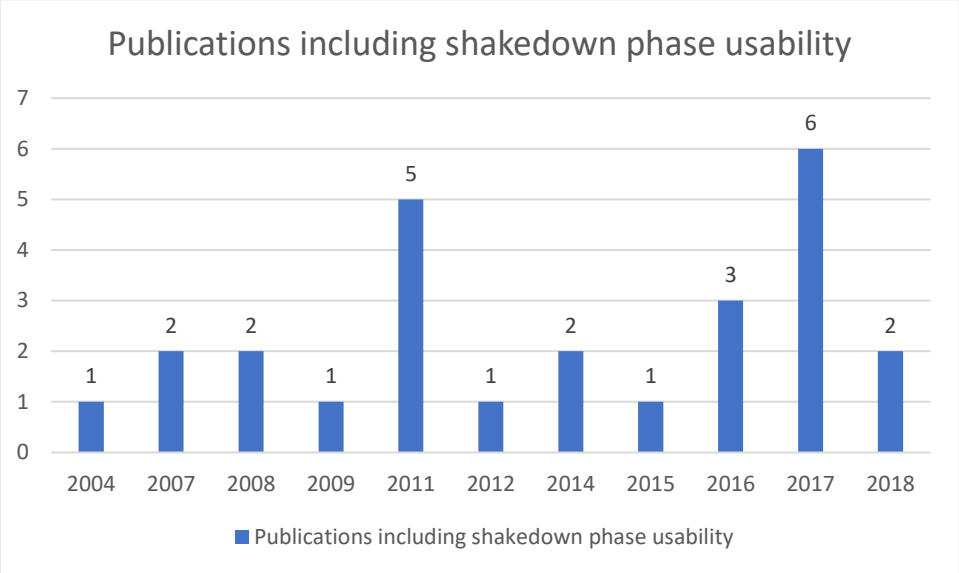

Figure 9. Publications including shakedown phase usability.

Onward and upward usability usage was also identified. These four papers that included Onward and upward stage usability used other usability methods in all the other stages too as it can be observed from the Appendix A. Table 6 includes the publications that included Onward and upward usability.

Table 6. Publications including Onward and upward phase usability.

First author	Year	Title	Type
Kato C.	2011	Development of online counseling system and usability evaluation	Article
Humayoun S.R.	2011	A three-fold integration framework to incorporate user-centered design into agile software development	Conference paper
Sohaib O.	2011	Incorporating discount usability in extreme programming	Article
Forbrig P.	2016	Continuous requirements engineering and human-centered agile software development	Conference paper

From Table 6 we can see that the only papers that included usability in Onward and upward stage are written within this decade. Also, two of which are published as article and two are published as conference paper. Three out of four titles also suggest that they are covering either integration or combination of methods from agile and from usability groups.

In Appendix A systematic overview of classification scheme is presented with a list of 92 publications included in this study that were deemed relevant. From the Appendix A it can be seen that 21 publications that included either of the above later stages' usability were listed as Conference papers by SCOPUS and only 5 listed as Articles. Also, in overall from the relevant publications most were listed as Conference papers.

Table 7. Papers including Shakedown or Onward and upward usability.

First author	Year	Source title
Jokela T.	2004	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
Memmel T.	2007	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
Memmel T.	2007	People and Computers XXI HCI.But Not as We Know It - Proceedings of HCI 2007: The 21st British HCI Group Annual Conference
Sain Z.H.	2008	Proceedings of the 1st International Conference on Advances in Computer-Human Interaction, ACHI 2008
Wolkerstorfer P.	2008	Conference on Human Factors in Computing Systems - Proceedings
Budwig M.	2009	Conference on Human Factors in Computing Systems - Proceedings
Kato C.	2011	Journal of Emerging Technologies in Web Intelligence
Humayoun S.R.	2011	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
Gonçalves J.	2011	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
Sadasivam R.S.	2011	Journal of Medical Internet Research
Sohaib O.	2011	International Journal of Software Engineering and its Applications
Hussain Z..	2012	ACHI 2012 - 5th International Conference on Advances in Computer-Human Interactions
Butt S.M.	2014	2014 International Conference on Computer and Information Sciences, ICCOINS 2014 - A Conference of World Engineering, Science and Technology Congress, ESTCON 2014 - Proceedings
Heimgärtner R.	2014	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
Rojas L.A.	2015	ACM International Conference Proceeding Series
Kropp E.	2016	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
Forbrig P.	2016	CEUR Workshop Proceedings
Sfetsos P.	2016	IISA 2016 - 7th International Conference on Information, Intelligence, Systems and Applications
Sekar B.	2017	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
Rahayu P.	2017	2016 International Conference on Information Technology Systems and Innovation, ICITSI 2016 - Proceedings
Daraghmi Y.-A.	2017	ACM International Conference Proceeding Series
South H.	2017	ICMI 2017 - Proceedings of the 19th ACM International Conference on Multimodal Interaction
Teka D.	2017	Proceedings of the IEEE International Conference on Computing, Networking and Informatics, ICCNI 2017
Terminanto A.	2017	IOP Conference Series: Materials Science and Engineering
Teka D.	2018	Proceedings - International Conference on Software Engineering
Rezaei-hachesu P.	2018	International Journal of Medical Informatics

In Table 7 there are publications listed that included either Shakedown usability or Onward and Upward usability. Table 7 shows that only six papers are published prior 2010. Which shows that even though no explicit timeframe was established during research the most papers that included usability in these later stages are published within the current decade. This can be seen as increased interest in usability in agile environment.

7. Discussion and implications

In this chapter discussion of results the implications are presented.

RQ1: How many papers include usability within agile software development? Results showed that in total of 92 papers included some sort of usability method used during the agile software development. Amount of papers found can be considered rather small when comparing the results to prevalence of agile development currently in software development.

Results imply that in raw comparison between search results, usability in general agile software development is maybe used in some way during the development, but it is not prevalent in the current literature. Even though the results of this study are based on the existence of the word "usability" in the paper and do not acknowledge usage of only specific usability method technique names. This is also supported by systematic review conducted by Salvador et al. (2014) were their initial search came up with 307 results, ultimately only 32 were selected to the study.

RQ2: At what stages of development were the usability methods applied? Study used classification scheme to establish overview of the field of study. Classification scheme consisted of dividing the development in four stages: Chartering, Project, Shakedown, Onward and upward. Study concluded that in 93% of papers usability method was used in Chartering stage. Project stage usability was found from 92% of papers. Papers that had the stage of usability usage identified 35% included Shakedown phase usability and only 5% of papers included Onward and upward phase usability.

Results gained from this study are not aligned with the results gained from systematic review published by Salvador et al. (2014). Their study suggested that only 22% used usability methods in requirements stage against 93% in this study in Chartering stage. Study conducted by Salvador et al. also had only 32 papers in their pool of papers included to last stage of the study. More than two times less than in SMS conducted in this thesis. This shows that choosing certain type of papers can have huge impact on the results of the study.

In this study the Project stage usability covers large area extending from end of requirements to first implementation which can be considered to be the biggest part of development. Comparing to study by Salvador et al. (2014) we can consider that Project phase included both design phase and implementation phase. In this area, their study is close to aligning with study conducted in this thesis. Their results suggested 50% had used usability methods in implementation phase and 40% in design phase.

RQ3: To which extent is usability mentioned in the later stages of the development? From the results it can be seen that papers were stage was identified from, 35% had Shakedown phase usage and 5% papers included Onward and upward phase usability. From this it can be concluded that since all the papers that included Onward and upward usability also included Shakedown phase usability that in conclusion 35% of papers mention usability in the later stages of the development.

From 75 papers only 35% included usability methods used in the later stages of development. When less than half of agile development projects decide to use any usability methods after initial implementation of system, it can be concluded that there is either something misunderstood from principles of usability or they are seen as non-

profitable, unnecessary addition. Therefore, this research suggests that more research is forwarded towards effects of these usability methods used in the later stages.

RQ: How is usability in agile software development noted in literature?

Currently, interest towards usability method usage in agile software development can be seen risen. Multiple researchers and studies have been conducted to integrate usability to agile methods from which few are included in this thesis (Brhel et al., 2015; Magues et al., 2017; Singh, 2008; Sohaib & Khan, 2010). This suggests that there is increasing interest towards this field of study. As did this study conclude the increased interest, so did Magues et al. (2017) in their study.

At the same times as, there is increase in interest, there can be seen gaps already in current form of study. As there have been multiple studies combining methods, but not so many on the effects of each method. Both agile and usability as their own are widely researched and developing new methods constantly. Still the effects of certain methods within the agile environment is not widely explained. Also, not all the agile environments are same so which effects are most effective in said context?

Usage of any usability methods in any stage of development has a positive effect on the outcome of the system. Still usability should be seen as an iterative tool, it should be applied and considered in every step of the way. Especially in the later stages, when system is in so called shakedown phase, where the system is implemented and there are still bug fixing and evaluation and redesigning the system to fit the real world needs of the system. Changes made during this stage in current literature are listed, but not discussed or acknowledged their effects of these changes. Fixes made should not just be solutions that work, but solutions that do not break the usability of the system.

8. Conclusion

In this thesis, presentation of the findings of SMS conducted on usability usage in agile development environment is given. The study was conducted using SMS method were papers were retrieved from SCOPUS database. Usage of SMS instead of SLR allowed to increase the number of papers could be taken in to study. The goal of the study was to look in to how usability in agile environment is presented in current literature with focus on later stages of usability. Initially 269 papers were chosen to include to the study, but usage of inclusion and exclusion criteria narrowed the amount of papers to include 92 relevant papers. From these 92 papers it was possible to identify stage from 75 papers.

Results showed that 93% of papers included Chartering stage usability, 92% included Project stage usability, 35% included shakedown stage usability and only 5% of papers included Onward and upward stage usability. Study also concluded that in 35% of papers included some sort of usability method used in the later stages of development.

Overall, this study contributes to knowledge by providing an overview of current literature about usability usage in agile software development. Consensus in the literature seems to be that usability usage in agile environment is still trying to find its place but there is research being done constantly to make it more prevalent in the field. From those agile software development projects, that had included the usage of some sort of usability method to development, too few included usability throughout the whole development cycle.

Considering the limitations of the study there are few that must be mentioned. One clear limitation was the usage of only one language (English) for the papers. Even though English is well established language used in research, there is possibility that there has been several publications and papers that are not included in this study based on the language barrier. Second limitation that was discovered during the study was that the identification of stage of usability was done with only one reviewer. This leaves room for bias and can affect the outcome of this study by few papers based on the fact that someone else could have either identified the stage from the paper or that in their opinion the stage was not clearly enough presented, or the paper could have been deemed irrelevant for the study.

Limiting factor that could be identified from the study was the usage of single database. Even when the amount of the papers was enough, the context and bias of several publication towards the subject could influence the study results. Usage of multiple databases could have increased the amount of papers for this study, but it would have also pushed the limitation of researcher since only one researcher was conducting the study. Lastly, limitation inherent in the SMS could have effect to this study too. Since SMS can be less accurate compared to SLR in terms of accuracy based on the fact that early cutting of papers was done by using only keywords, titles and abstract instead of full text. The usage of SMS could have had negative effect on the study, but it also enables the study to be more comprehensive overview of the field of focus and allowed creating clear image of the field instead of narrow, more focused view. It also allows the researchers to decide themselves if and what future research work needs to be focused on.

Future work within this field should focus more on giving usability more comprehensive usage and to consider using usability methods in other stages of development to gain maximum effect from it. There is research already done when considering joining the usability methods towards the agile development, but as it can be seen from the results,

most of these are published in conference proceedings. Also, more comprehensive research could be interesting to eliminate some limitations of this study.

References

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New directions on agile methods: A comparative analysis. Paper presented at the *25th International Conference on Software Engineering, 2003,* 244-254.

Beck, K. (1999). Embracing change with extreme programming. *Computer,* (10), 70-77.

Beck, K., & Gamma, E. (2000). *Extreme programming explained: Embrace change* addison-wesley professional.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. & (Thomas, D. (2001). Manifesto for agile software development. Retrieved from http://agilemanifesto.org/principles.html

Benington, H. D. (1956). Production of large computer programs. Paper presented at the *ONR Symposium on Advanced Programming Methods for Digital Computers, June 1956, Pp 15-27.* 15-27.

Bevan, N. (2001). International standards for HCI and usability. *International Journal of Human-Computer Studies, 55*(4), 533-552.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer,* (5), 61-72.

Brhel, M., Meth, H., Maedche, A., & Werder, K. (2015). Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology, 61*, 163-181.

Flavián, C., Guinalíu, M., & Gurrea, R. (2006). The role played by perceived usability, satisfaction and consumer trust on website loyalty. *Information & Management, 43*(1), 1-14.

Gould, J. D., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communications of the ACM, 28*(3), 300-311. doi:10.1145/3166.3170

Hassenzahl, M., Beu, A., & Burmester, M. (2001). Engineering joy. *IEEE Software, 18*(1), 70-76.

Hertzum, M. (2010). Images of usability. *Intl.Journal of Human–Computer Interaction, 26*(6), 567-600.

International Organization for Standardization. (1991). *Software engineering - product quality.* (No. ISO/IEC 9126:1991). Retrieved from https://www.iso.org/standard/16722.html

International Organization for Standardization. (2001). *Software engineering - product quality - part 1: Quality model.* (No. ISO/IEC 9126-1:2001). Retrieved from https://www.iso.org/standard/22749.html

International Organization for Standardization. (2018). *Ergonomics of human-system interaction - part 11: Usability: Definitions and concepts.* (No. ISO 9241-11:2018). Retrieved from https://www.iso.org/standard/63500.html

Jeffries, R., Anderson, A., & Hendrickson, C. (2001). *Extreme programming installed* Addison-Wesley Professional.

Kitchenham, B. A., Budgen, D., & Brereton, P. O. (2011). *Using mapping studies as the basis for further research – A participant-observer case study* doi:https://doi.org/10.1016/j.infsof.2010.12.011

Magues, D. A., Castro, J. W., & Acuna, S. T. (2017). Usability in agile development: A systematic mapping study. Paper presented at the doi:10.1109/CLEI.2016.7833347 Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-85013870278&doi=10.1109%2fCLEI.2016.7833347&partnerID=40&md5=d1770655a1815f542aa660d08597a34c

Markus, M. L., & Tanis, C. (2000). The enterprise systems experience-from adoption to success. *Framing the Domains of IT Research: Glimpsing the Future through the Past,* 173-207.

Nah, F. F., Lau, J. L., & Kuang, J. (2001). Critical factors for successful implementation of enterprise systems. *Business Process Management Journal, 7*(3), 285-296.

Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM, 50*(3), 79-83. doi:10.1145/1226736.1226739

Nielsen, J. (1994). *Usability engineering* Elsevier.

Paulk, M. C. (2001). Extreme programming from a CMM perspective. *IEEE Software, 18*(6), 19-26.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. Paper presented at the *Ease, 8* 68-77.

Petersen, K., Wohlin, C., & Baca, D. (2009). The waterfall model in large-scale development. Paper presented at the *International Conference on Product-Focused Software Process Improvement,* 386-400.

Raccoon, L. (1997). Fifty years of progress in software engineering. *ACM SIGSOFT Software Engineering Notes, 22*(1), 88-104.

Rising, L., & Janoff, N. S. (2000). The scrum software development process for small teams. *IEEE Software, 17*(4), 26-32.

Royce, W. W. (1970). Managing the development of large software systems: Concepts and techniques. Paper presented at the *IEEE WESCOM. IEEE Computer Society Press, Los Alamitos (1970),*

Ruparelia, N. B. (2010). Software development lifecycle models. *SIGSOFT Softw.Eng.Notes, 35*(3), 8-13. doi:10.1145/1764810.1764814

Salvador, C., Nakasone, A., & Pow-Sang, J. A. (2014). A systematic review of usability techniques in agile methodologies. Paper presented at the *Proceedings of the 7th Euro American Conference on Telematics and Information Systems,* 17.

Schwaber, K. (1997). Scrum development process. *Business object design and implementation* (pp. 117-134) Springer.

Scott, J. E., & Vessey, I. (2002). Managing risks in enterprise systems implementations. *Communications of the ACM, 45*(4), 74-81. doi:10.1145/505248.505249

Shneiderman, B. (2000). Universal usability. *Communications of the ACM, 43*(5), 84-91. doi:10.1145/332833.332843

Singh, M. (2008). U-SCRUM: An agile methodology for promoting usability. Paper presented at the *Agile 2008 Conference,* 555-560. doi:10.1109/Agile.2008.33

Sohaib, O., & Khan, K. (2010). Integrating usability engineering and agile software development: A literature review. Paper presented at the *2010 International Conference on Computer Design and Applications, 2,* V2-32-V2-38.

Sonderegger, A., & Sauer, J. (2010). The influence of design aesthetics in usability testing: Effects on user performance and perceived usability. *Applied Ergonomics, 41*(3), 403-410.

Sutherland, J., & Schwaber, K. (2017). The scrum guide. Retrieved from http://scrumguides.org/scrum-guide.html

Vanderheiden, G. (2000). Fundamental principles and priority setting for universal usability. Paper presented at the *Proceedings on the 2000 Conference on Universal Usability,* 32-37.

Vuksanovic, I. P., & Sudarevic, B. (2011). Use of web application frameworks in the development of small applications. Paper presented at the *2011 Proceedings of the 34th International Convention MIPRO,* 458-462.

Wells, D. (2017). Extreme programming (XP). Retrieved from https://www.agilealliance.org/glossary/xp/

Appendix A. Systematic map overview using classification scheme.

First author	Year	Cited by	Document Type	Chartering	Project	Shakedown	Onward and upward	Can't be identified
Becker S.A.	2002	11	Article	X	X			
Kane D.	2003	35	Conference Paper					X
Jokela T.	2004	25	Article	X	X	X		
Schatz B.	2005	72	Article					X
Hodgetts P.	2005	16	Conference Paper	X	X			
Hurlbutt T.	2006	2	Conference Paper	X				
Lievesley M.A.	2006	6	Conference Paper	X	X			
Lee J.C.	2006	21	Conference Paper	X	X			
Memmel T.	2007	8	Conference Paper	X	X	X		
Ferreira J.	2007	6	Conference Paper	X	X			
Parsons D.	2007	6	Conference Paper	X	X			
Memmel T.	2007	53	Conference Paper	X	X	X		
Düchting M.	2007	25	Conference Paper	X	X			
Haikara J.	2007	24	Conference Paper	X	X			
Memmel T.	2007	21	Conference Paper	X	X			
Lee J.C.	2007	41	Conference Paper	X	X			
Sain Z.H.	2008	15	Conference Paper	X	X	X		
Singh M.	2008	59	Conference Paper	X	X			
Dubinsky Y.	2008		Conference Paper	X	X			
Hussain Z.	2008	22	Conference Paper	X				
Sy D., Miller L.	2008	25	Conference Paper					X
Obendorf H.	2008	29	Conference Paper	X	X			
Wolkerstorfer P.	2008	32	Conference Paper	X	X	X		
Bonacin R.	2009	8	Conference Paper	X	X			
Hussain Z.	2009	26	Conference Paper	X	X			
Wusteman J.	2009	6	Article	X	X			
Budwig M.	2009	27	Conference Paper	X	X	X		
Adikari S.	2009	21	Conference Paper	X	X			
Fisher K.G.	2009	4	Conference Paper	X	X			
Peixoto C.S.A.	2009	4	Conference Paper					X
Hussain Z.	2009	15	Conference Paper					X
Peixoto C.S.A.	2009	6	Conference Paper	X	X			
Jiménez-Mixco V.	2009	4	Conference Paper					X
Srinivas S.	2010		Conference Paper					X
Gershon R.	2010	83	Article					X
Ghanam Y.	2010	16	Conference Paper					X
Kato C.	2011	4	Article	X	X	X	X	
Humayoun S.R.	2011	22	Conference Paper	X	X	X	X	

Gonçalves J.	2011	6	Conference Paper	X	X	X		
Realinho V.	2011	5	Conference Paper	X	X			
Sadasivam R.S.	2011	16	Article	X	X	X		
Jeon S.	2011	13	Conference Paper	X	X			
Sohaib O.	2011	14	Article	X	X	X	X	
Faulring A.	2012	5	Conference Paper					X
Luján-Mora S.	2012	4	Conference Paper	X	X			
Lárusdóttir M.K.	2012	4	Conference Paper					X
Dehling T.	2012	5	Conference Paper	X				
Hussain Z.	2012	7	Conference Paper	X	X	X		
Grigoreanu V.	2013	15	Conference Paper	X	X			
Raison C.	2013	9	Conference Paper	X	X			
Maguire M.	2013	9	Conference Paper	X	X			
Wan Ahmad W.F.	2013		Conference Paper	X	X			
Bourimi M.	2013	1	Conference Paper					X
Cajander Å.	2013	20	Conference Paper					X
González C.S.	2013	2	Conference Paper	X	X			
Butt S.M.	2014	2	Conference Paper	X	X	X		
Wale-Kolade A.Y.	2014	6	Conference Paper	X	X			
Isa W.A.R.W.M.	2014	2	Conference Paper	X	X			
Lizano F.	2014	6	Conference Paper	X	X			
Chammas A.	2014		Conference Paper					X
Heimgärtner R.	2014	1	Conference Paper	X	X	X		
Krusche S.	2014	9	Conference Paper	X	X			
González-González C.S.	2015		Article	X	X			
Ahmad W.F.W.	2015		Conference Paper	X				
Wale-Kolade A.Y.	2015	9	Article	X	X			
Branham C.	2015	1	Conference Paper	X	X			
Rojas L.A.	2015	1	Conference Paper	X	X	X		
de Freitas R.C.	2016		Conference Paper	X	X			
Mesa A.	2016		Conference Paper		X			
Forbrig P.	2016	1	Conference Paper	X	X			
Kropp E.	2016		Conference Paper	X	X	X		
Forbrig P.	2016		Conference Paper	X	X	X	X	
Wale-Kolade A.	2016	3	Article					X
Bhaskar R.K.	2016	1	Conference Paper	X	X			
Sfetsos P.	2016	3	Conference Paper	X	X	X		
Sekar B.	2017		Conference Paper	X	X	X		
Merino E.	2017		Conference Paper	X	X			
Fontdevila D.	2017	1	Conference Paper					X
Daly C.	2017		Conference Paper	X	X			
Rahayu P.	2017	1	Conference Paper	X	X	X		
Mutiawani V.	2017	1	Conference Paper		X			
Rico-Olarte C.	2017	2	Article	X	X			
Daraghmi Y.-A.	2017		Conference Paper	X	X	X		
Kifle M.	2017	2	Conference Paper	X	X			

South H.	2017		Conference Paper	X	X	X		
Teka D.	2017		Conference Paper	X	X	X		
Terminanto A.	2017	1	Conference Paper			X		
Sensuse D.I.	2018		Conference Paper	X	X			
Teka D.	2018		Conference Paper	X	X	X		
Pastor D.	2018		Conference Paper					X
Fuertes W.	2018		Conference Paper		X			
Rezaei-hachesu P.	2018		Article			X		

Appendix B. List of papers used in creating systematic overview

Authors	Title	Year	Source title	Cited by	Document Type
Becker S.A., Berkemeyer A.	Rapid application design and testing of Web usability	2002	IEEE Multimedia	11	Article
Kane D.	Finding a place for discount usability engineering in agile development: Throwing down the gauntlet	2003	Proceedings of the Agile Development Conference, ADC 2003	35	Conference Paper
Jokela T., Abrahamsson P.	Usability assessment of an extreme programming project: Close co-operation with the customer does not equal to good usability	2004	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	25	Article
Schatz B., Abdelshafi I.	Primavera gets Agile: A successful transition to Agile development	2005	IEEE Software	72	Article
Hodgetts P.	Experiences integrating sophisticated user experience design practices into Agile processes	2005	Proceedings - AGILE Confernce 2005	16	Conference Paper
Hurlbutt T., Klemmer S.R.	Bridging the gap: Fluidly connecting paper notecards with digital representations for story/task-based planning	2006	Conference on Human Factors in Computing Systems - Proceedings	2	Conference Paper
Lievesley M.A., Yee J.S.R.	The role of the interaction designer in an agile software development process	2006	Conference on Human Factors in Computing Systems - Proceedings	6	Conference Paper
Lee J.C.	Embracing agile development of usable software systems	2006	Conference on Human Factors in Computing Systems - Proceedings	21	Conference Paper
Memmel T., Gundelsweiler F., Reiterer H.	CRUISER: A cross-discipline user interface and software engineering lifecycle	2007	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	8	Conference Paper

Ferreira J., Noble J., Biddle R.	Interaction designers on eXtreme Programming teams: Two case studies from the real world	2007	Proceedings of NZCSRSC 2007, the 5th New Zealand Computer Science Research Student Conference	6	Conference Paper
Parsons D., Lal R., Ryu H., Lange M.	Software development methodologies, agile development and usability engineering	2007	ACIS 2007 Proceedings - 18th Australasian Conference on Information Systems	6	Conference Paper
Memmel T., Gundelsweiler F., Reiterer H.	Agile human-centered software engineering	2007	People and Computers XXI HCI.But Not as We Know It - Proceedings of HCI 2007: The 21st British HCI Group Annual Conference	53	Conference Paper
Düchting M., Zimmermann D., Nebe K.	Incorporating user centered requirement engineering into agile software development	2007	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	25	Conference Paper
Haikara J.	Usability in agile software development: Extending the interaction design process with personas approach	2007	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	24	Conference Paper
Memmel T., Reiterer H., Holzinger A.	Agile methods and visual specification in software development: A chance to ensure universal access	2007	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	21	Conference Paper
Lee J.C., McCrickard D.S.	Towards extreme(ly) usable software: Exploring tensions between usability and agile software development	2007	Proceedings - AGILE 2007	41	Conference Paper
Sain Z.H., Lechner M., Milchrahm H., Shahzad S., Slaný W., Umgeher M., Vlk T., Wolkerstorfer P.	User interface design for a mobile multimedia application: An iterative approach	2008	Proceedings of the 1st International Conference on Advances in Computer-Human Interaction, ACHI 2008	15	Conference Paper

Singh M.	U-SCRUM: An agile methodology for promoting usability	2008	Proceedings - Agile 2008 Conference	59	Conference Paper
Dubinsky Y., Humayoun S.R., Catarci T.	Eclipse plug-in to manage user centered design	2008	CEUR Workshop Proceedings		Conference Paper
Hussain Z., Lechner M., Milchrahm H., Shahzad S., Slany W., Umgeher M., Wolkerstorfer P.	Agile user-centered design applied to a mobile multimedia streaming application	2008	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	22	Conference Paper
Sy D., Miller L.	Optimizing Agile user-centred design	2008	Conference on Human Factors in Computing Systems - Proceedings	25	Conference Paper
Obendorf H., Finck M.	Scenario-based usability engineering techniques in agile development processes	2008	Conference on Human Factors in Computing Systems - Proceedings	29	Conference Paper
Wolkerstorfer P., Tscheligi M., Sefelin R., Milchrahm H., Hussain Z., Lechner M., Shahzad S.	Probing an agile usability process	2008	Conference on Human Factors in Computing Systems - Proceedings	32	Conference Paper
Bonacin R., Baranauskas M.E.C., Rodrigues M.A.	An agile process model for inclusive software development	2009	Lecture Notes in Business Information Processing	8	Conference Paper
Hussain Z., Milchrahm H., Shahzad S., Slany W., Tscheligi M., Wolkerstorfer P.	Integration of extreme programming and user-centered design: Lessons learned	2009	Lecture Notes in Business Information Processing	26	Conference Paper
Wusteman J.	OJAX: A case study in agile Web 2.0 open source development	2009	Aslib Proceedings	6	Article
Budwig M., Jeong S., Kelkar K.	When user experience met agile: A case study	2009	Conference on Human Factors in Computing Systems - Proceedings	27	Conference Paper

Adikari S., McDonald C., Campbell J.	Little design up-front: A design science approach to integrating usability into agile requirements engineering	2009	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	21	Conference Paper
Fisher K.G., Bankston A.	From extreme programming and usability engineering to extreme usability in software engineering education (XP+UE→XU)	2009	Proceedings - 2009 Agile Conference, AGILE 2009	4	Conference Paper
Peixoto C.S.A.	Human-computer interface expert system for agile methods	2009	Proceedings of the International Conference on Information Technology Interfaces, ITI	4	Conference Paper
Hussain Z., Slany W., Holzinger A.	Investigating agile user-centered design in practice: A grounded theory perspective	2009	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	15	Conference Paper
Peixoto C.S.A., Da Silva A.E.A.	A conceptual knowledge base representation for agile design of human-computer interface	2009	3rd International Symposium on Intelligent Information Technology Application, IITA 2009	6	Conference Paper
Jiménez-Mixco V., Villalar González J.L., Arca A., Cabrera-Umpierrez M.F., Arredondo M.T., Manchado P., García-Robledo M.	Application of virtual reality technologies in rapid development and assessment of ambient assisted living environments	2009	1st ACM SIGMM International Workshop on Media Studies and Implementations that Help Improving Access to Disabled Users, MSIADU'09, Co-located with the 2009 ACM International Conf. Multimedia,MM'09	4	Conference Paper
Srinivas S., Biswas A., Srinivasan J.	An application synopsis tool for database applications developed using oracle application express	2010	ISEC'10 - Proceedings of the 2010 India Software Engineering Conference		Conference Paper

Gershon R., Rothrock N.E., Hanrahan R.T., Jansky L.J., Harniss M., Riley W.	The development of a clinical outcomes survey research application: Assessment centerSM	2010	Quality of Life Research	83	Article
Ghanam Y., Andreychuk D., Maurer F.	Reactive variability management in agile software development	2010	Proceedings - 2010 Agile Conference, AGILE 2010	16	Conference Paper
Kato C., Shiono Y., Goto T., Tsuchida K.	Development of online counseling system and usability evaluation	2011	Journal of Emerging Technologies in Web Intelligence	4	Article
Humayoun S.R., Dubinsky Y., Catarci T.	A three-fold integration framework to incorporate user-centered design into agile software development	2011	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	22	Conference Paper
Gonçalves J., Santos C.	POLVO - Software for prototyping of low-fidelity interfaces in agile development	2011	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	6	Conference Paper
Realinho V., Dias A.E., Romão T.	Testing the usability of a platform for rapid development of mobile context-aware applications	2011	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	5	Conference Paper
Sadasivam R.S., Delaughter K., Crenshaw K., Sobko H.J., Williams J.H., Coley H.L., Ray M.N., Ford D.E., Allison J.J., Houston T.K.	Development of an interactive, web-delivered system to increase provider-patient engagement in smoking cessation	2011	Journal of Medical Internet Research	16	Article
Jeon S., Han M., Lee E., Lee K.	Quality attribute driven agile development	2011	Proceedings - 2011 9th International Conference on Software Engineering Research, Management and Applications, SERA 2011	13	Conference Paper

Sohaib O., Khan K.	Incorporating discount usability in extreme programming	2011	International Journal of Software Engineering and its Applications	14	Article
Faulring A., Myers B.A., Oren Y., Rotenberg K.	A case study of using HCI methods to improve tools for programmers	2012	2012 5th International Workshop on Co-operative and Human Aspects of Software Engineering, CHASE 2012 - Proceedings	5	Conference Paper
Luján-Mora S., Masri F.	Integration of Web accessibility into agile methods	2012	ICEIS 2012 - Proceedings of the 14th International Conference on Enterprise Information Systems	4	Conference Paper
Lárusdóttir M.K., Cajander Å., Gulliksen J.	The big picture of UX is missing in scrum projects	2012	CEUR Workshop Proceedings	4	Conference Paper
Dehling T., Sunyaev A.	Architecture and design of a patient-friendly eHealth web application: Patient information leaflets and supplementary services	2012	18th Americas Conference on Information Systems 2012, AMCIS 2012	5	Conference Paper
Hussain Z., Lechner M., Milchrahm H., Shahzad S., Slany W., Umgeher M., Vlk T., Ḱoffel C., Tscheligi M., Wolkerstorfer P.	Practical usability in XP software development processes	2012	ACHI 2012 - 5th International Conference on Advances in Computer-Human Interactions	7	Conference Paper
Grigoreanu V., Mohanna M.	Informal Cognitive Walkthrough (ICW): Paring down and pairing up for an agile world	2013	Conference on Human Factors in Computing Systems - Proceedings	15	Conference Paper
Raison C., Schmidt S.	Keeping user centred design (UCD) alive and well in your organisation: Taking an agile approach	2013	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	9	Conference Paper

Maguire M.	Using human factors standards to support user experience and agile design	2013	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	9	Conference Paper
Bourimi M., Kesdogan D.	Experiences by using AFFINE for building collaborative applications for online communities	2013	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	1	Conference Paper
Cajander Å., Larusdottir M., Gulliksen J.	Existing but not explicit - The user perspective in scrum projects in practice	2013	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	20	Conference Paper
González C.S., Toledo P., Muñoz V., Noda M.A., Bruno A., Moreno L.	Inclusive educational software design with agile approach	2013	ACM International Conference Proceeding Series	2	Conference Paper
Butt S.M., Ahmad W.F.W., Rahim L.	Handling tradeoffs between agile and usability methods	2014	2014 International Conference on Computer and Information Sciences, ICCOINS 2014 - A Conference of World Engineering, Science and Technology Congress, ESTCON 2014 - Proceedings	2	Conference Paper
Wale-Kolade A.Y., Nielsen P.A., Päivärinta T.	Integrating usability practices into agile development: A case study	2014	Information Systems Development: Transforming Organisations and Society Through Information Systems - Proceedings of the 23rd International Conference on Information Systems Development, ISD 2014	6	Conference Paper

Isa W.A.R.W.M., Lokman A.M., Aris S.R.S., Aziz M.A., Taslim J., Manaf M., Sulaiman R.	Engineering rural informatics using agile user centered design	2014	2014 2nd International Conference on Information and Communication Technology, ICoICT 2014	2	Conference Paper
Lizano F., Sandoval M.M., Stage J.	Integrating usability evaluations into Scrum: A case study based on remote synchronous user testing	2014	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	6	Conference Paper
Chammas A., Quaresma M., Mont'Alvão C.R.	An analysis of design methodologies of interactive system for mobiles	2014	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)		Conference Paper
Heimgärtner R., Solanki A.	Using agile methods in intercultural HCI design projects	2014	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	1	Conference Paper
Krusche S., Bruegge B.	User feedback in mobile development	2014	MobileDeLi 2014 - Proceedings of the 2nd International Workshop on Mobile Development Lifecycle, Part of SPLASH 2014	9	Conference Paper
González-González C.S., Toledo-Delgado P., Muñoz-Cruz V.	Agile human centered methodologies to develop educational software [Metodologías ágiles centradas en personas para desarrollar software educativo]	2015	DYNA (Colombia)		Article
Ahmad W.F.W., Azahari I.N.N.	Visual approach of a mobile application for autistic children: Little routine	2015	Communications in Computer and Information Science		Conference Paper

Wale-Kolade A.Y.	Integrating usability work into a large inter-organisational agile development project: Tactics developed by usability designers	2015	Journal of Systems and Software	9	Article
Branham C., Moxley J., Ross V.	My reviewers: Participatory design & crowd-sourced usability processes	2015	SIGDOC 2015 - Proceedings of the 33rd Annual International Conference on the Design of Communication	1	Conference Paper
Rojas L.A., Macías J.A.	An agile information-architecture-driven approach for the development of user-centered interactive software	2015	ACM International Conference Proceeding Series	1	Conference Paper
de Freitas R.C., Rodrigues L.A., Jr., da Cunha A.M.	AGILUS: A method for integrating usability evaluations on agile software development	2016	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)		Conference Paper
Mesa A., Largo S.M., Nañola C., Agrazamendez M., Novero A.	A web-based system for marine fishes mapping and assessment	2016	37th Asian Conference on Remote Sensing, ACRS 2016		Conference Paper
Forbrig P., Saurin M.	Supporting the HCI aspect of agile software development by tool support for UI-pattern transformations	2016	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	1	Conference Paper
Kropp E., Koischwitz K.	Experiences with user-centered design and agile requirements engineering in fixed-price projects	2016	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)		Conference Paper
Forbrig P.	Continuous requirements engineering and human-centered agile software development	2016	CEUR Workshop Proceedings		Conference Paper
Wale-Kolade A., Nielsen P.A.	Apathy Towards the Integration of Usability Work: A Case of System Justification	2016	Interacting with Computers	3	Article

Bhaskar R.K., Anslow C., Brosz J., Maurer F.	Developing usable APIs with XP and cognitive dimensions	2016	Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC	1	Conference Paper
Sfetsos P., Angelis L., Stamelos I., Raptis P.	Integrating user-centered design practices into agile Web development: A case study	2016	IISA 2016 - 7th International Conference on Information, Intelligence, Systems and Applications	3	Conference Paper
Sekar B.	Enterprise software experience design: Journey and lessons	2017	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)		Conference Paper
Merino E., Zapata C., Aguilar M.C.	UCD and agile methodology in the development of a cultural heritage platform	2017	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)		Conference Paper
Fontdevila D., Genero M., Oliveros A.	Towards a usability model for software development process and practice	2017	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	1	Conference Paper
Daly C., Zapata C., Paz F.	Improving the usability in a video game through continuous usability evaluations	2017	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)		Conference Paper
Rahayu P., Sensuse D.I., Fitriani W.R., Nurrohmah I., Mauliadi R., Rochman H.N.	Applying usability testing to improving Scrum methodology in develop assistant information system	2017	2016 International Conference on Information Technology Systems and Innovation, ICITSI 2016 - Proceedings	1	Conference Paper

Mutiawani V., Subianto M., Tony H.R.	A web-based agricultural commodity price information system for Aceh region, Indonesia	2017	Proceedings - 2016 12th International Conference on Mathematics, Statistics, and Their Applications, ICMSA 2016: In Conjunction with the 6th Annual International Conference of Syiah Kuala University	1	Conference Paper
Rico-Olarte C., López D.M., Narváez S., Farinango C.D., Pharow P.S.	Haphop-physio: A computer game to support cognitive therapies in children	2017	Psychology Research and Behavior Management	2	Article
Daraghmi Y.-A., Daadoo M., Eleyan D.	User centered method for enhancing the adoption of software systems in Palestine	2017	ACM International Conference Proceeding Series		Conference Paper
Kifle M., Dittrich Y., Teka D.	Contextualizing user centered design with agile methods in Ethiopia	2017	2017 IEEE AFRICON: Science, Technology and Innovation for Africa, AFRICON 2017	2	Conference Paper
South H., Taylor M., Dogan H., Jiang N.	Digitising a medical clerking system with multimodal interaction support	2017	ICMI 2017 - Proceedings of the 19th ACM International Conference on Multimodal Interaction		Conference Paper
Teka D., Dittrich Y., Kifle M.	Integrating discount usability in scrum development process in Ethiopia	2017	Proceedings of the IEEE International Conference on Computing, Networking and Informatics, ICCNI 2017		Conference Paper
Terminanto A., Hidayat R., Hidayanto A.N.	Implementation of enterprise resource planning using Odoo module sales and CRM. Case study: PT Ecosains Hayati	2017	IOP Conference Series: Materials Science and Engineering	1	Conference Paper
Sensuse D.I., Satria D., Pratama A.A., Wulandari I.A., Mishbah M., Noprisson H.	Integrating UCD into Scrumban for better and faster usability design	2018	2017 International Conference on Information Technology Systems and Innovation, ICITSI 2017 - Proceedings		Conference Paper

Teka D., Dittrich Y., Kifle M.	Adapting lightweight user-centered design with the scrum-based development process	2018	Proceedings - International Conference on Software Engineering		Conference Paper
Pastor D., Arcos-Medina G., Onate A., Loaiza M., Torres J.	Semantic Query System for Moodle Virtual Courses Based on an Ontology	2018	2018 5th International Conference on eDemocracy and eGovernment, ICEDEG 2018		Conference Paper
Fuertes W., Tunala A., Moncayo R., Meneses F., Toulkeridis T.	Software-Based Platform for Education and Training of DDoS Attacks Using Virtual Networks	2018	Proceedings - 2017 International Conference on Software Security and Assurance, ICSSA 2017		Conference Paper
Rezaei-hachesu P., Samad-Soltani T., Yaghoubi S., GhaziSaeedi M., Mirnia K., Masoumi-Asl H., Safdari R.	The design and evaluation of an antimicrobial resistance surveillance system for neonatal intensive care units in Iran	2018	International Journal of Medical Informatics		Article
Wan Ahmad W.F., Butt S.M., Rahim L.	Usability evaluation of the agile software process	2013	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)		Conference Paper