

Juha Hänninen

**PROMOTING USABILITY IN SOFTWARE DEVELOPMENT SCRUM PROJECTS**

Bachelor's thesis



UNIVERSITY OF JYVÄSKYLÄ  
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS  
2015

## ABSTRACT

Hänninen, Juha

Promoting usability in software development Scrum projects

University of Jyväskylä, 2015, 22 p.

Information Systems, Bachelor's Thesis

Supervisor: Rousi, Rebekah

Usability is often taken for granted in software development and it does not always get the attention it deserves. The benefits of usability are often unclear for the developers, which makes it difficult for them to understand its meaning. Especially in agile Scrum projects, where time is short, usability can be seen as something that can be excluded, since it is not seen as important.

This thesis examines the role of usability in software development Scrum projects and the methods which can be used to promote usability's role. The thesis is a literature review explaining these methods without taking a stand on their suitability on different kinds of use contexts.

The object of this thesis is to give an idea about how usability can be improved in the software development Scrum projects by applying different methods. These results can be used in development projects in practice, although the use contexts should be further researched.

Keywords: Usability, user-centered design, user-centered agile, Scrum, U-SCRUM

# TIIVISTELMÄ

Hänninen, Juha

Käytettävyyden edistäminen ohjelmistokehityksen Scrum-projekteissa

Jyväskylän yliopisto, 2015, 22 s.

Tietojärjestelmätiede, kandidaatin tutkielma

Ohjaaja: Rousi, Rebekah

Käytettävyyttä pidetään usein itsestäänselvyyttenä ohjelmistokehityksessä eikä se saa sen vuoksi aina tarvitsemaansa huomiota. Käytettävyyden hyödyt ovat kehittäjille usein epäselviä, minkä vuoksi sen merkitystä on vaikea ymmärtää. Erityisesti ketterissä Scrum-projekteissa, joissa aikaa on rajoitetusti, käytettävyys saatetaan helposti jättää pois koska se ei tunnu tärkeältä.

Tämä tutkielma on kirjallisuuskatsaus, jossa tarkastellaan käytettävyyden huomioon ottamista suunnittelussa, käyttäjien huomioimista, käytettävyyden roolia ohjelmistokehityksen Scrum-projekteissa sekä erilaisia metodeja käytettävyyden aseman parantamiseksi. Metodien sopivuuteen eri käyttöympäristöissä ei oteta kantaa, koska tätä ei ole vielä tutkittu tarpeeksi.

Tämän tutkielman tarkoitus on selittää kuinka käytettävyyden asemaa voidaan parantaa ohjelmistokehityksen Scrum-projekteissa käyttämällä mainittuja metodeja. Tutkielman tuloksia voi ottaa käytäntöön ohjelmistokehitysprojekteissa. Eri metodien sopivuutta erilaisiin käyttötilanteisiin tulisi kuitenkin tutkia lisää.

Asiasanat: käytettävyys, käyttäjäkeskeinen suunnittelu, käyttäjäkeskeinen ketterä menetelmä, Scrum, U-SCRUM

# CONTENT

ABSTRACT .....	2
TIIVISTELMÄ .....	3
1 INTRODUCTION .....	5
2 USABILITY IN DESIGN.....	7
2.1 User Involvement .....	9
2.2 User Consideration in Design.....	10
3 SCRUM .....	11
4 PROMOTING USABILITY IN SCRUM PROJECTS.....	13
4.1 U-SCRUM .....	14
4.2 User-Centered Design.....	15
4.3 User-Centered Agile.....	16
4.3.1 The UCA Process.....	17
4.3.2 The Effectiveness of UCA .....	18
5 CONCLUSION .....	19
REFERENCES.....	21

# 1 INTRODUCTION

Usability is often neglected in software development. There are many reasons for this, such as lack of time and underestimation of its importance. Usability is also often taken for granted and not enough resources are allocated to deal with usability issues (Gould & Lewis, 1985). Because of this, there is a need to promote usability, especially in software development Scrum projects where tight schedules are often a problem. When deadlines are approaching and something must be left out, usability is easy to ignore since its meaning is not understood well enough. Therefore, usability must be integrated into the development so that it is not a separate part of the development. Instead, it must be present at all times and in all development activities. According to Göransson et al. (2003), usability has not been yet integrated seamlessly into software development, which is why developers should try to focus more on usability and extensive user involvement.

Usability is the cornerstone of software. Without good usability users are not able to achieve their goals with the software, even if it would work well otherwise. One might also say that if software is not usable, it does not actually work properly. Because of this, usability requires a lot of attention in the development phase and also even before the actual development. In this thesis the following research questions are answered:

- How can usability be promoted in software development Scrum projects?
- What methods have been developed for this?

These questions are answered by first analyzing usability's role in design and representing three methods: U-SCRUM, User-Centered Design, and User-Centered Agile. These methods can be used in software development, when usability is considered as important and there is a need to ensure its position.

This thesis has been made as a literature review using scientific articles from the Google Scholar database as well as conference papers and books relating to the subject. The publishing year of the sources used in this thesis vary from 1985 to 2015. It is inevitable that changes have happened during this time

in the field of usability but some of the old rules still apply. The main focus of this thesis is on explaining the different methods that help to improve usability and the conclusions made from the research.

The first chapter focuses on usability's role in design. Also user involvement and user consideration are discussed. After this, the agile project management method Scrum is explained. Then an overall view of the integration of Scrum and usability methods is given, after which the three different methods, U-SCRUM, User-Centered Design and User-Centered Agile are examined. The thesis ends with a conclusion of the methods discussed and the role of usability in Scrum projects. Finally, some topics for further research are considered.

## 2 USABILITY IN DESIGN

Usability design is an important part of software development in order to create a product, which meets the demands of the user. In order to understand usability design we must first define “usability”. Nielsen (1993) defines usability as a combination of multiple components, which are learnability, efficiency, memorability, errors and satisfaction. The ISO 9241-11 (1998) standard defines usability as the following:

*“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.”*

Another definition of usability was made by Göransson, Gulliksen and Boivie (2003). They emphasize active user involvement through the development process in user-centered design (UCD). User involvement will be discussed further in chapter 2.1 and UCD in chapter 4.2. Göransson et al. (2003) summarize the usability design process as being a tool for integrating usability aspects into software development.

Taking usability principles into consideration in development projects is considered to be obvious, although often these principles are not used (Gould & Lewis, 1985). The developers might also have a hard time imagining the problems the real users could face, if they never encounter difficulties themselves. To address these problems, Gould and Lewis (1985) suggest having direct contact with the end-user already before the actual system design. Nevertheless, developers might consciously depreciate the importance of usability, when it is not considered to be a purchase factor (van Kuijk, van Driel & van Eijk, 2015). Additionally, increased usability tends to save costs, but these savings may not be visible until the final release of the product (Nielsen, 1993).

Gould and Lewis (1985) suggest three usability principles to be utilized in system design:

1. Early focus on users and tasks
2. Empirical measurement
3. Iterative design

The first principle encourages the designers to recognize the users and how they use the product. The second principle suggests using simulations and prototypes in order to acquire information about the actual work process with the product. The last principle says that design should be iterative so that when problems are found, they will be handled in the next cycle of the design. This iteration process should last as long as necessary.

In usability engineering product development is made with the help of customer data and feedback. This information can be acquired by direct observation and interaction with the user. This produces more reliable data compared to self-reporting of the user. (Sohaib & Khan, 2010.) However, usability engineering suffers often from the variation of the techniques in user analysis, usability goal specification and design evaluation. Usability engineering also does not address the whole development process. Instead, different usability engineering techniques are added to the development process separately (Göransson et al., 2003). Göransson et al. (2003) also worry about the fact that the descriptions and applications of these techniques are not necessarily appropriate for the overall development process. This can cause, for example, the use of irrelevant data. According to them, the successful integration of usability in software development requires a process perspective, so that the development process is seen as a whole. This way usability is not seen separate from the development. Furthermore, a usability professional should take part in the development at all times, instead of occasionally consulting. Göransson et al. (2003) also acknowledge the fact that usability does not get the attention it would need in software development since it is often taken for granted.

Another problem is that usability is not understood very well. Further, the users' participation and their experience of usability are thought to be subjective. This causes some developers to think that usability design comes along with uncertainty. Traditionally, usability has not been integrated into other activities in system development (Cajander, 2010.) Cajander (2010) additionally mentions that in development projects there are not usually many usability specialists and they seem to easily fall behind their tight schedules. The customer who has ordered the system is often blamed for the lack of usability if they have not included usability as a quality criterion for accepting the final product (Göransson et al., 2003).

According to Göransson et al. (2003), usability design in software development consists of four stages. These stages are: 1) early and continual focus on users, 2) evaluation with users, 3) iterative design and 4) integrated design. As mentioned before, Göransson et al. (2003) also mention that it is important to remember to include usability specialist(s) throughout the development process and all these stages.

Furthermore, Cooper (1999) recognizes the problem that developers seldom are designing for the users in particular. Instead, their focus is on the features. Features themselves do not offer any extra value to a user, who only wants to achieve the goal he or she is using the software for. Different features may be needed to achieve these goals, although they might also distract the us-



er from the essential activity. (Cooper, 1999.) According to Nielsen (1993), the simplicity of an interface ensures usability, unlike having every possible feature included. The more features and items of information found from an interface, the more it requires learning, increasing the risk of misunderstanding. When the simplicity is reduced it also takes more time for the user to find the intended feature. (Nielsen, 1993.)

Another problem Cooper (1999) mentions is that instead of ease-of-use, programs are designed so that users have to learn how to use them. However, ease-of-use and ease of learning are usually substitutes for each other (Mayhew, 1999). Thus, learnability is also one of the usability factors (Nielsen, 1993). According to Nielsen (1993) the first experience users get from using a system is the learning. Users do not usually learn the interface completely prior to the use. Therefore, the learnability while using is an important aspect of the overall usability. (Nielsen, 1993.) In general, when designing user interfaces, the developers have to make a compromise between powerful functionality and simplicity (Mayhew, 1999).

## 2.1 User Involvement

The main benefit of user involvement in the development is that it assures the suitability of the product for the intended use in the environment where it will be used (Abrams, Maloney-Krichmar, & Preece, 2004). Abrams et al. claim that this results in more effective, efficient and safe products. User involvement can happen in various ways. In some development projects the users may only be consulted about their needs, observed and allowed to participate in usability testing. On the other hand, users might also participate as partners with the designers throughout the design process. (Abrams et al., 2004.)

Göransson et al. (2003) remind about the effect that user involvement has on software development. Although user involvement is often considered resource demanding, there are no shortcuts in user involvement and it usually enables savings in the long run.

The relationship between the developers and the end-users should be interactive so that the end-users would be a part of the actual design team (Gould & Lewis, 1985). Even though users should be included in the design team, Gould and Lewis (1985) remind that the difference between the designers and the end-users must remain. Otherwise, users might become too aware of the technical problems and will not be able to examine the software objectively. However, it should be noticed that including users in the design process does not necessarily mean improved usability (Boivie, Åborg, Persson & Löfberg, 2003).

An important aspect of user involvement is that the users testing the product have to represent the intended users. It is also necessary to plan for the involvement from the first day of the project. Using concrete design representations and simple terminology in the design process is important, so that the par-

ticipating end-users can more easily be involved in the design. (Gulliksen, Göransson, Boivie, Blomkvist, Persson, & Cajander, 2003.)

## 2.2 User Consideration in Design

Designers should be aware of the intended users' characteristics, which might affect their skills or imply of possible learning difficulties. These characteristics include for example, work experience, educational level, age and previous experience with computers. (Nielsen, 1993.) Nielsen also points out that the developers probably cannot imagine all possible scenarios where the users make misinterpretations of interface elements or perform actions in a totally different way than what was intended. Therefore, user understanding is essential. Gulliksen et al. (2003) stress the fact that the focus should be on the users throughout the entire system lifecycle.

Users do not usually complain about a system being able to do too many actions. However, this does not mean that all included features are important or necessary. Therefore, it is important for the developers to be aware of the usability problems, which the extra features can cause. (Nielsen, 1993.) Peters, Dieckmann, Dixon, Hibbard and Mertz (2007) noticed that when presenting a lot of information at the same time, it burdens the user's cognitive skills and reduces the ability to make good decisions.

When designing user interfaces, the designers should think of what users understand. User interfaces should not include technical terms, instead every action should be explained with terms the user understands. (Nielsen, 1993.) Nielsen (1993) also suggests that the user interface should be consistent, so that every command and action always have the same effect. According to Nielsen it helps the user to trust to the system and allows the user to explore the system even further.

Nielsen (1993) suggests that the system should give feedback for the user to help the user understand what is happening when actions are performed. Nielsen proposes that also positive feedback is given when actions have been completed successfully instead of just informing about errors, which should be avoided entirely if possible.

Nielsen (1993) mentions localization as an important aspect of usability. If a system is used in many different countries or regions, symbols and actions might have a different meaning for the users. Therefore it is important to make localized versions of these sections. According to Shneiderman and Plaisant (2005) the cultural differences also affect the preferences and the abilities users have. Furthermore they mention that localization requires much more than just translated terms, for example characters, numerals and even cultural conventions and etiquette.

### 3 SCRUM

Scrum is an agile method for project management, where the product development process is iterative and adaptive. Scrum is based on iterations, which are known as sprints. These sprints have a fixed length and the activities conducted in a sprint remain the same throughout the whole development process. A new sprint is started immediately after the previous has been completed. (Deuff & Cosquer, 2013.) In the beginning of each sprint the planning is made and a review is given in the end (Hossain, Babar & Paik, 2009).

When the duration of a development project is unexpectedly extended, it often causes the failure of the project. It is also one of the main reasons for project failures. Therefore, agile methods were developed so that software development could be faster when time is short. The whole idea of agile methods is to make the whole development process simpler and to serve the customer better. (Sohaib & Khan, 2010.) In Scrum, a usable product is created after each sprint, which makes it possible to deliver the final product after any sprint. When a product is released another sprint can be made in order to make possible changes based on customer feedback and the financial success of the product. (Rising & Janoff, 2000.)

The customer plays an important role with multiple responsibilities in agile software development. The participation of the customer in the development should concentrate on user stories and testing. Collaboration with the customer might however conflict with the needs of the end-user, since the customer does not necessarily represent the typical user. The customer ordering the product is not necessarily aware of any usability demands the product has. Therefore, it is important for the developer to know and recognize the users. (Sohaib & Khan, 2010.) Göransson et al. (2003) remind that face-to-face communication between the project members and the users is crucial, since agile developers favor communication centric approach in development projects.

The three roles of Scrum include the product owner, the Scrum master and the development team. The product owner takes care of defining the product content, manages priorities of the product, and makes sure that the development team understands these priorities. The Scrum master is responsible for the

application of the Scrum method in the project and helps the development team to adapt the method. The development team is responsible for the product development while organizing itself trying to optimize productivity, flexibility, creativity and to enhance its skills. (Deuff & Cosquer, 2013.)

In agile software development individuals and interactions are usually valued over processes and tools. Additionally, a working software package has a higher priority than comprehensive documentation. The interaction with the customer is more collaboration than negotiation. Agile methods also enable adapting to change when necessary, instead of sticking to the original plan. (Sy, 2007.) Sy (2007) argues in favor of agile user-centered design, since it enables more product design. Agile methods in general also seem to result in better user experience than waterfall projects, where the design is its own phase and is made entirely before the development and testing. In agile projects usability is investigated throughout the project development. Designs are prioritized so that no time is wasted for unused designs. (Sy, 2007.) Scrum-method suits best projects where it is difficult or impossible to define requirements in advance and where uncertainty is expected throughout the product development (Rising & Janoff, 2000).

## 4 PROMOTING USABILITY IN SCRUM PROJECTS

The need to promote usability already in the development process springs from the evolution of software development. The human-computer interaction (HCI) field and software engineering have been separated into their own paths, which has led to a situation where both parties have knowledge of their own area, but not of the other (Göransson et al., 2003).

Göransson et al. (2003) suggest that usability professionals should learn more about software development in order to understand the possibilities and limitations of the development tools better. Moreover, software developers should become familiar with usability and user-centered systems design. Since the HCI field is still relatively young, many software developers do not have any experience in it (Göransson et al., 2003). Van der Veer and van Vliet (2003) argue that user interface design should be considered from the very beginning of the development project. They emphasize the fact that it requires the basic knowledge of HCI. Therefore it is important that every developer is aware of the basics of HCI.

Because HCI and software engineering seem to have their own specialists they do not usually have much to do with each other. This is why Göransson et al. (2003) suggest that it would be important to bring these fields closer to one another. Later I examine three different possibilities to promote usability in Scrum projects: Usability Scrum (U-SCRUM), User-Centered Design (UCD) and User-Centered Agile (UCA).

The focus of usability is on the end users' work with the software. The focus of agile development is on the way of developing the software. Combining these two still seems to cause problems to developers. While software developers are succeeding in agile software development, these projects rarely involve any usability engineering. Software might be usable for an expert, but not to a novice. Therefore, it is important to remember that agile methods alone do not ensure usability of software, even though agile methods have similarities to UCD. There are also some problems when trying to integrate these two, such as communication issues and time differences between the designers and devel-

opers. Further, the amount of user involvement might vary a lot. (Sohaib & Khan, 2010.)

Göransson et al. (2003) consider that the focus in agile development should be shifted towards design, even when user analyses, usability requirements and evaluation results remain an important input to the design. They also propose usability tools, techniques and methods to be integrated in the software development process.

When a usability specialist is assigned to a project, he usually works close to the user and takes part in analyzing usability. The results from these analyses are then transferred into design activities. The usability specialist continues participating in prototype design and takes part in following evaluation activities. This way iteration is achieved. (Göransson & Sandbäck, 1999.) Iterations help to make small changes to the design which can be evaluated by the user before the final design is chosen (Gulliksen et al., 2003). Since usability design can also be iterative, it implies that usability design could naturally fit into agile development. Usability design also tends to have similarities to U-SCRUM, which was introduced by Singh (2008). According to Göransson and Sandbäck (1999), the usability designer should be an active participator in the design and development instead of being just another project manager. Singh (2008) additionally suggests having two product owners in U-SCRUM. U-SCRUM is discussed later in chapter 3.1.

One problem that Scrum product owners often face is the pressure of marketing and sales which can complicate the focus on usability. Too often product owners are not skilled and motivated enough, which would be necessary in order to design effective user experiences. (Singh, 2008.) Singh also says that traditional agile methodologies do not adequately enable usability design, which is why she proposes U-SCRUM to be used as an agile method, in order to take usability more into consideration. U-SCRUM is discussed further in the next chapter.

In the integration of agile methods and UCD, the focus should be divided equally on usability evaluation and design. User personas and product prototypes can be used to help the design. (Da Silva, Martin, Maurer & Silveira, 2011.) Da Silva et al. (2011) also mention that not enough controlled experiments have been made about integrating UCD and agile methods. Therefore, more empirical research would be needed on this field.

## 4.1 U-SCRUM

U-SCRUM is an agile software development method. It was designed to incur higher usability in products compared to traditional Scrum projects. In U-SCRUM, a development project has two product owners instead of the traditional one. One product owner focuses on usability and user experience, and the other is responsible for the more traditional functions in the development. (Singh, 2008.)

According to Singh (2008), the reason for developing the U-SCRUM methodology is mainly the basic limitation of traditional Scrum. The focus and motivator has been perceived to be the completion of the product instead of enhanced user experience. Even in projects where usability specialists have been assigned, usability's role seems to diminish when tight on schedule. Singh also mentions, that a working product is accepted too often without being properly tested and designed with a usability focus. The problem still seems to be the lack of experience of developers, which is why they do not have the skills to approach the implementation of tasks from a user-centric perspective.

Some limitations for U-SCRUM still remain. Having two product owners, it is vital that they remain peers. Otherwise, it is possible that the arrangement tends to change toward a conventional Scrum project. Also worse problems could occur, such as if the two product owners begin to challenge each other's position as a product owner. Having two product owners may require additional arrangements in the project, although according to Singh's experience, this is not usually a major factor since all team members are usually highly engaged in the project.

Singh noticed that U-SCRUM can be best utilized when the products developed are complex and novel. In projects where the topic is well understood, the benefit of two product owners reduces. In familiar use contexts the needs of the user are probably well known and the extra coordination of U-SCRUM would just result in additional costs. However, U-SCRUM should be further studied in different use-contexts to confirm this.

## 4.2 User-Centered Design

In user-centered design (UCD), user involvement is utilized to improve usability through the understanding of the user and task requirements. The design and evaluation is usually iterative throughout the process. UCD is commonly accepted as a key part in product usefulness and usability. That is why UCD is considered to effectively overcome the limitations of traditional system-centered design. (Mao, Vredenburg, Smith & Carey, 2005.) In UCD it is important to have general knowledge about the user and their needs, while still having focus on usability. The developers should also be aware of the aspects of HCI and the research on the field. (Deuff & Cosquer, 2013.)

Lately it has been noticed that user involvement in UCD can be most beneficial when the findings are effectively communicated. Therefore, it is important to further consider the manner of communication instead of just focusing on the methods of user involvement. (Van Kuijk et al., 2015.)

Göransson et al. (2003) presented a special UCD case, where UCD was considered to be a process focusing on usability throughout the whole development process. UCD remained an important part of the project until the end of the system life cycle. In this case, UCD was based on user-centeredness and involvement, usability skills within the team, iteration, continuous evaluation and

specific design activities. According to Göransson et al. (2003), clients usually only care about the solution the product is serving as. Therefore, it is important to focus on providing that solution, which is the design, e.g. a word processor or a photo editor.

It is also mentionable that UCD is not an issue only in software development: Mao et al. (2005) emphasize the fact that UCD has become more increasingly important also outside software development. The growing popularity of e-commerce has its own demands for usability, since users want to be able to make purchases online with just a few clicks. Mao et al. (2005) also found it surprising, that UCD was often considered as time consuming, even though it has been realized to cause savings in time and money in the long run. The savings are based on the reduction of rework which is needed with unusable products. Mao et al. (2005) however, express some concerns about UCD. The effectiveness of UCD is often measured with a variation of techniques. Further, it was not very clear to the development teams whether UCD had actually caused any savings in development time and costs. UCD in general seems to be a blurry compilation of techniques and methods without any real standards and common practices. According to Mao et al. (2005), UCD is mainly focusing on the total user experience, end-to-end user involvement during the development and measuring customer satisfaction.

According to Iivari & Iivari (2006) taking every individual users' capabilities into consideration and fully satisfying each individual user's needs is the ideal of UCD. The system should support the unique activities of each user as well as their preferences and characteristics. However, problems may occur in the design because of the diversity of the users. It might be difficult to include every type of user in the development or to take their needs into consideration. Users also learn all the time while using the system, which makes it more difficult for the developers to evaluate the actual usability. If the system allows personalization to some extent, it may ease the usability issues some users have, when their individual needs have not been thoroughly examined. (Iivari & Iivari, 2006.) However, Nielsen (1993) points out that especially novice users do not use personalization features even if they are made possible. Therefore, Nielsen recommends that the system should support novice users with the features instead of personalization.

According to Deuff and Cosquer (2013), the most difficult aspect of combining UCD and agile methods is the design before the actual development phase. From an agile perspective it is no use spending too much time on the original design since alterations are inevitable during the iterations. Otherwise the pace of the iterations along with agility would suffer.

### **4.3 User-Centered Agile**

The user-centered agile method (UCA) was introduced by Deuff and Cosquer (2013) to integrate the agile Scrum-method to UCD. Deuff and Cosquer mention



that UCA is a compromise, where UCD and agile methods have been adjusted for better compatibility. This way the working framework of the project is balanced and shared within the team. The UCA method has three phases: 1) the design phase, 2) the development phase, and 3) the validation phase. The design phase yields a global view of the product and is a compilation of UCD and the pre-sprint phase of Scrum. The development phase continues the application of UCD with mini-rounds of user testing, which should be conducted regularly. The last phase is the validation phase where the traditional user testing should take place. Deuff and Cosquer also remind that the user experience (UX) team should adapt to the pace which the agile process requires. Therefore, there is limited time before the actual development stage begins.

According to Deuff and Cosquer (2013) most developers do not reserve much time for research about the users and their use context before the development, even it is a crucial phase. This research could also be done outside the agile period, when it would add more value to the project. Deuff and Cosquer highlight the fact that UX experts should be given enough time to gather data from real users in order to comprehend the whole project and functional objectives. This way the UX specialist can better understand the requirements of their work. It helps sharing the workload with the rest of the team and supports prioritization with knowledge of the users and sets out objectives with the product owner.

In UCA the UX experts can form their own team or be a part of the development team. However, it is important to take care of proper communication when the UX experts have a team of their own. When integrated to other development teams communication tends to be sufficient. Being in the same team also helps the whole team share the values which usually underline the role of usability, which helps collaboration because the work begins much before the development. Still it should be kept in mind that the UX experts must follow the rules and adjust to the pace of the developers in order to avoid falling behind the original schedule. (Deuff & Cosquer, 2013.)

Deuff and Cosquer (2013) remind of the importance of teamwork. To improve the teamwork, they propose the tasks proposed in the sprint review to be done by the whole agile team. They also recommend all of the project members to regularly follow the mini-user testing sessions, which helps understanding the problems users face.

#### **4.3.1 The UCA Process**

The first phase of UCA is a preparation phase prior to the actual development. During the pre-sprint, the development team creates user stories in the product backlog. In this phase the UX designer tries to find out who the real users are, their needs, and the use context of the upcoming product. The intention of this pre-sprint phase is to implement the UCD method by the UX designer. This is done with design workshops which can include the product owner team as well as the agile team. (Deuff & Cosquer, 2013.)

In the second phase the actual development starts. The product, which was designed in the first phase is created. The design is then perfected during the development phase with the feedback received from PO team and end-users.

The third and last phase is the validation phase. Usually a traditional user-testing round is scheduled in the validation phase. This requires the PO to budget an evaluation ergonomist who is responsible of offering support to the developers and of the necessary changes caused by user feedback. Technical support for the end-users is also usually required during this phase.

### **4.3.2 The Effectiveness of UCA**

Deuff and Cosquer (2013) explain that UCD will not stop at any stage of the development thanks to the UCA method. This results from the continuous presence of the UX professionals, which include ergonomists, designers and UX designers. UCD is further enforced with user involvement at every stage of the development. UCA tries to disable the possibility of having the same person designing and evaluating the product, which would result in a conflict of interest. Furthermore, in the UCA method, a UX designer is assigned also to the PO team. Having a separate designer in the PO team and an ergonomist in the development team improves communication and handling usability issues. (Deuff & Cosquer, 2013.)

## 5 CONCLUSION

In this thesis I examined how the position of usability could be improved in software development Scrum projects. First, the usability's role in design was examined. Then user involvement and user consideration were discussed. In the second chapter the agile development method Scrum was explained. The last chapter covered the three methods for promoting usability in Scrum projects, including U-SCRUM, User-Centered Design and User-Centered Agile.

When trying to include usability in agile development projects, it is important to include users in the development as well as having usability professionals. Another key point is that all of the development team members should have at least adequate knowledge about usability. It should be noticed that agile and user-centered methods usually fit naturally together. In user-centered methods iterations are used to get feedback even from the slightest changes. Also testing is made in every iteration. This enables including usability methods to agile projects. The problem is fitting the usability and agile iterations to each other. Another aspect concerning the integration of usability and agile methods is the time allocation. There has to be compromises on how much time is reserved for design, without harming the iterations and pace of agile methods.

There has not been comprehensive study concerning how these different methods would apply on different development projects. However, according to Singh (2008), U-SCRUM is best suitable for projects where the topic is novel and not well understood. UCD is a compilation of usability improving methods, which usually include user involvement and thorough testing. UCA is another agile usability-promoting development method. Further research about the use context of UCD and UCA is required, in order to further understand their suitability for different development projects. Especially the UCA method would need some confirmation, since it has not been widely studied.

Furthermore, it should be remembered that usability as a concept might change over time. What we now perceive as usable might not apply a decade later. For example, user interfaces change largely as well as the devices in rather short periods of time. This is precisely what has happened with mobile devices and user interfaces. Therefore, old rules about usable user interfaces are not

necessarily sufficient anymore, or must be at least examined critically. The role of usability will not diminish in the future, since all kinds of electronic services are increasingly becoming inevitable for normal users, such as online bank services. This results in the need of constantly focusing on users rather than the system. These aspects must be considered in the future research as well as the different user types.

As a conclusion there has not been definite understanding of promoting usability in software development Scrum projects. However, the different methods presented all seem to emphasize the role of usability specialists in charge as well as the usability knowledge of all development team members. User involvement throughout the development process is also required to further understand the underlying issues within the product.

## REFERENCES

- Abras, C., Maloney-Krichmar, D., & Preece, J. (2004). User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, 37(4), 445-456.
- Boivie, I., Åborg, C., Persson, J., & Löfberg, M. (2003). Why usability gets lost or usability in in-house software development. *Interacting with Computers*, 15(4), 623-639.
- Cajander, Å. (2010). *Usability – Who cares?: The introduction of user-centred systems design in organisations*. Doctoral dissertation. Uppsala: Acta Universitatis Upsaliensis.
- Cooper, A. (1999). *The inmates are running the asylum: [Why high-tech products drive us crazy and how to restore the sanity]* (Vol. 261). Indianapolis: Sams.
- Da Silva, T. S., Martin, A., Maurer, F., & Silveira, M. S. (2011). User-centered design and agile methods: A systematic review. In *AGILE* (pp. 77-86).
- Deuff, D., & Cosquer, M. (2013). *User-centered Agile Method*. John Wiley & Sons.
- Gould, J. & Lewis, C. (1985). Designing for usability : Key principles and what designers think. *Communications of the ACM*, 28(3), 300-311
- Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., & Cajander, Å. (2003). Key principles for user-centred systems design. *Behaviour and Information Technology*, 22(6), 397-409.
- Göransson, B., Gulliksen, J. & Boivie, I. (2003). The usability design process – Integrating user-centered systems design in the software development process. *Software Process Improvement and Practice*, 8, 111-131.
- Göransson, B., & Sandbäck, T. (1999). Usability designers improve the user-centred design process. In *Proceedings for INTERACT* (Vol. 99, pp. 1-4).
- Hossain, E., Babar, M. A., & Paik, H. Y. (2009). Using scrum in global software development: a systematic literature review. In *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on* (pp. 175-184). IEEE.
- Iivari, J., & Iivari, N. (2006). Varieties of user-centeredness. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on* (Vol. 8, pp. 176a-176a). IEEE.

- ISO/IEC. 9241-11. (1998). Ergonomic requirements for office work with visual display terminals (VDT)s - Part 11: Guidance on usability, ISO/IEC 9241-11: 1998.
- Mao, J. Y., Vredenburg, K., Smith, P. W., & Carey, T. (2005). The state of user-centered design practice. *Communications of the ACM*, 48(3), 105-109.
- Mayhew, D. J. (1999). The usability engineering lifecycle. In *CHI'99 Extended Abstracts on Human Factors in Computing Systems* (pp. 147-148). ACM.
- Nielsen, J. (1993). Usability Engineering. *Elsevier*.
- Peters, E., Dieckmann, N., Dixon, A., Hibbard, J. H., & Mertz, C. K. (2007). Less is more in presenting quality information to consumers. *Medical Care Research and Review*, 64(2), 169-190.
- Rising, L., & Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE software*, 17(4), 26-32.
- Shneiderman, S. B., & Plaisant, C. (2005). Designing the user interface 4 th edition. ed: *Pearson Addison Wesley, USA*.
- Singh, M. (2008). U-SCRUM: An agile methodology for promoting usability. In *Agile, 2008. AGILE'08. Conference* (pp. 555-560). IEEE.
- Sohaib, O., & Khan, K. (2010). Integrating usability engineering and agile software development: A literature review. In *Computer design and applications (ICCDA), 2010 international conference on* (Vol. 2, pp. V2-32). IEEE.
- Sy, D. (2007). Adapting usability investigations for agile user-centered design. *Journal of usability Studies*, 2(3), 112-132.
- van der Veer, G., & Van Vliet, H. (2003). A plea for a poor man's HCI component in software engineering and computer science curricula; after all: The human-computer interface is the system. *Computer science education*, 13(3), 207-225.
- van Kuijk, J., van Driel, L. & van Eijk, D. (2015). Usability in product development practice; an exploratory case study comparing four markets. *Applied Ergonomics*, 47, 308-323.