Chapter 10
Challenges from Integrating Usability Activities in Scrum: Why Is Scrum so Fashionable?

Marta Lárusdóttir, Åsa Cajander, Gudbjörg Erlingsdottir, Thomas Lind, and Jan Gulliksen

Abstract Scrum is currently a widely used process in most areas of software development. Conversely, usability activities as prescribed in the area of HCI are not widely used in the software industry, especially not in agile software development projects. Through an analysis of interview and survey data from five studies we scrutinize the reasons for choosing Scrum, consequences of using Scrum, and study the challenges of integrating usability activities in Scrum projects are scrutinized. Our results show that the IT professionals appreciate the inherent values in Scrum, which are speed and communication internal to the Scrum team. Also, working in teams and focusing on a small number of tasks at a time is valued. The main challenges are that including specialists in the teams is hard and Scrum does not always match with external requirements for the organizations. Usability activities in Scrum are found to be informal and implicit, even sometimes hidden behind more fashionable concepts such as security and accessibility to increase priority. In addition, usability activities are often seen as not fitting in the pace of the project. Two of the underlying questions in the paper are: Why is Scrum so fashionable? How can usability activities be better integrated in agile projects? Answers to these questions are discussed in the chapter.

M. Lárusdóttir (✉)
School of Computer Science, Reykjavik University, Reykjavik, Iceland
e-mail: marta@ru.is

Å. Cajander • T. Lind
Department of Information Technology, Uppsala University, Uppsala, Sweden
e-mail: asa.cajander@it.uu.se; thomas.lind@it.uu.se

G. Erlingsdottir
Lund University, Lund, Sweden
e-mail: gudbjorg.erlingsdottir@design.lth.se

J. Gulliksen
KTH Royal Institute of Technology, Stockholm, Sweden
e-mail: gulliksen@kth.se

G. Cockton et al. (eds.), *Integrating User-Centred Design in Agile Development*,
Human–Computer Interaction Series, DOI 10.1007/978-3-319-32165-3_10

Keywords Agile software development • Scrum • Usability • User experience • User centred design • Fashion management

10.1 Introduction

Software development methods and processes have changed over the years, and one management idea adapted to software development has been followed by the other. Some of these processes focus on project management, whereas others include everything from business modelling to code structuring as for example the previously popular process the Rational Unified Process [1]. Some of the software development methods that have been popular are based on explicit core values such as communication or customer satisfaction that are emphasised in agile software development processes [2]. Since the launch of the Agile Manifesto in 2001 [3], agile methods such as Extreme Programming, Dynamic Systems Development Method and Kanban have gained popularity with Scrum as being the most popular process [4]. Scrum has indeed become so popular that it has expanded the domain of software development and moved into other areas. The ideas and core values of Scrum are hence integrated areas far beyond software engineering and computer science. Recently we have observed it being used as the silver bullet of supervision for PhD students, as well as in the instructional enterprise of teaching computer science. Here the ideas used are continuous feedback as well as planning ahead in sprints. One of our underlying questions in the paper is hence: Why is Scrum so fashionable?

From the survey that Version One launched in 2014 [4] one can see that the reasons given for using Scrum are: “Accelerate product delivery” (59 %), “Enhance ability to manage changing requirements” (56 %), “Increase productivity” (53 %) and “Enhance software quality” (46 %). When asked about how they measure the success of their agile initiatives, the most frequent measurements are: “On-time delivery” (58 %), “Product quality” (48 %) and “Customer/user satisfaction” (44 %). It is interesting to note that when asked about the benefits of using agile, extended customer/user satisfaction is not on the list. Moreover, when explicitly asking companies about why they use Scrum, we often also get the answer that this development process works better than the previous processes used, but also that Scrum has both advantages and disadvantages [5–7]. Still, there is truly a collective belief that using Scrum is a better way of working. It has been an implicit assumption about Scrum is that it addresses user perspectives better than traditional software processes [8], and that by simply applying an agile development process, systems could become usable for end-users. However, previous research has shown that this is not always the case, and that the context of Scrum impacts user involvement as described in, for example [5, 7]. Another underlying question in the paper is: How can usability activities be better integrated within agile projects?

There is still much to explore when it comes to the reasons for some methods to become popular, whereas others remain unknown. Hence, based on five empirical studies of Scrum and usability activities, we scrutinize the reasons for using Scrum, what the consequences of the choice were and what challenges were met when integrating usability activities in Scrum. Hence the main contribution of this paper is answering the following questions:

- What are the companies' reasons for using Scrum?
- What are the experienced consequences of using Scrum?
- What are the challenges of integrating usability activities in Scrum?

Finally, we also discuss the two underlying questions above. We discuss if one of the reasons for Scrum being so popular could be that it has become a management fashion, as defined by Abrahamson [9]:

"a relatively transitory collective belief, disseminated by management fashion setters, that a management technique leads rational management progress" (p. 257). Moreover, we discuss and relate the popularity of Scrum to the Diffusion of Innovation model by Rogers [10]. Additionally, we discuss how usability activities could be better integrated in Scrum projects.

The main contribution of the paper is a deeper understanding of the work environment that IT professionals working in Scrum projects have and the challenges they have when integrating usability activities in their software development.

10.2 Background

In this section we present the definition of the Scrum development process in the literature, how usability activities are explained in the literature and how these are practiced in the industry, the background of the integration of usability activities in Scrum in practice and finally the theories on fashion and diffusion of innovation.

10.2.1 Defining the Scrum Development Process

Scrum is currently a popular process used in most areas of software development [4]. In this subsection we will describe how the originators of Scrum define the process.

Jeff Sutherland and Ken Schwaber described the Scrum process in 1995 [2, 11]. They are the official pioneers of the process. In that document, Scrum is called a process. Today the process is owned by the Scrum Alliance, which certifies consultants and provides courses in Scrum. The paper that inspired Scrum had the title "The new new product development game" [12]. The authors explain "the rules of the game in new product development are changing. It takes more than high quality, low cost, and differentiation to excel in today's competitive market. It also takes speed and flexibility".

Some discussions have been whether Scrum is a framework or a process. Among the information that IT professionals ordinarily want to extract from descriptions of software development processes are information on: what is going to be done, when and where it will be done, how and why it will be done, who is going to do it, and who is dependent on it being done [13]. This is provided by Scrum, so we therefore call it a *process* in this paper.

Various types of software development processes have been suggested for the last 30 years. The main types are sequential software development, iterative development and incremental software development. The sequential software process is often named the Waterfall process [14]. The fundamental thought in this process is that it is sequential, so that each stage should be more or less finished before the next begins. In the 1990s more emphasis in software development was on delivering parts of the software to customers iteratively and incrementally, so time to market would be shorter [14]. In incremental software development the software requirements are divided into parts, which are implemented and a deliverable version of that part of the software is made [15]. The basic idea is that the IT professionals use the knowledge they gained in previous increments, both from developing the software and from users, to iterate the current version of the software during the development of current and future increments. The implementation starts with a simple set of requirements, and iteratively evolves until a full version of the software is implemented in a plan-driven way. This is also the fundamental idea behind Scrum.

In Scrum, the projects should be split up into 2–4 week long iterations called *sprints*, each aiming to end up with a potentially shippable product. In Scrum, self-organizing and strongly united teams are heavily emphasized, typically with six to eight interdisciplinary team members. One of the benefits of using agile development processes was claimed to be that customers' needs are taken more into account than when developing software using sequential processes [2]. Scrum is a simple process with a few ceremonies, roles and artefacts. The roles are called Product Owner, Scrum Master and Team Member [16]. The Scrum team is self-organising and works independently during the sprints. Daily Scrum meetings are prescribed where the Scrum team meets and plans the work during the day, and where the tasks are distributed in the group. The work in the Scrum team should be guided by collaboration and communication. Demos of the outcome are made at the end of every sprint.

The Product Owner has the responsibility to represent the needs and ideas of the customer, being responsible for writing the so called user stories and for managing these in a document called the Product Backlog. User stories are described in a standardised way to express the users' needs and the business value in sentences with a specific structure. The Product Backlog contains the requirements for the software to be built, often described as user stories. Parts of the Product Backlog may be technical and parts may be more human oriented. The project is often analysed to some extent and a vision of the product is made before the actual iterations start. That period is often referred to as Sprint Zero or pre-study phase.

The sprints are planned in sprint planning meetings, and the requirements to be addressed in each sprint are defined in the Sprint Backlog. By the end of each sprint, the Scrum team should release a potentially shippable product. At this point it is recommended in Scrum that the team collect feedback from the stakeholders to adapt the software to the users' needs in subsequent sprints. The team and the Product Owner define an exit-criterion of the items in the Sprint Backlog called "the definition of done". It describes when the items are completed and can be delivered to the customer. The definition might include descriptions of tests to be done. The Scrum board is a physical or electronic task board used by the team members to manage the different tasks and what to do during each sprint. The Scrum Master is often considered as a coach for the Scrum team. The Scrum Master is responsible for the process used during the software development and can, for example, decide how long the sprints should be.

10.2.2 Usability Activities

When computer development grew from being systems developed by operators for the operators themselves, to developing computer systems to be used by non-computer science users for the purpose of supporting their work tasks, the need to focus on usability became evident. The multidisciplinary research field of human computer interaction grew out of human factors and cognitive psychology, providing a theoretical framework for the study of humans interacting with systems [17].

The ISO 9241-210 standard [18] called: "Human-centred design for interactive systems", is a common framework for defining how to include the users' needs in software development in theory. Four major usability activities are described in the ISO 9241-210 [18] standard that shall take place during the development of software. First the usability activities should be planned. Then the four major four activities are explained: (a) Understand and specify the context of use, (b) Specify the user requirements, (c) Produce design solutions to meet these requirements, and (d) Evaluate the designs against requirements. It is stated in the standard that these four usability activities are interdependent and each activity uses outputs from the other activities. This means that it is recommended in the standard that when a specification of the context of use has been made, the user requirements could be described based on this understanding. Additionally, design solutions could be made based on these user requirement specifications. The outcomes of each of the three human-centred design activities should be evaluated and iterated according to the evaluation results, where appropriate. The ISO 9241-210 standard [18] defines the human-centred design as: "An approach to system design and development that aims to make interactive systems more usable by focusing on the use of the system, applying human factors/ergonomics and usability knowledge and techniques (p. 2)" It is also stated in the standard that: "Human-centred design activities can be incorporated in design approaches as diverse as object-oriented, waterfall, HFI (human factors integration), agile and rapid development (p. 10)." The usability activities should therefore have the capability to fit into the agile development process of Scrum as well as in any other development processes.

In the late twentieth century discussions of user experience (UX) broadened consideration of usage qualities beyond usability, and sought a concept to include more satisfaction-related qualities that were already associated with games and entertainment technologies [19]. UX considerations extended usability activities beyond quality-increasing testing and improvement to focus more on attractiveness (with the intention of increasing sales), and relating more to design than to testing [20]. UX has become a concept that people relate to outside of the immediate sector, meaning that marketing and project managers need to address these issues on a more strategic level [21]. Thus we now see interaction designers and UX lead roles at a managerial level [22]. All of a sudden there is money to be earned through proactive UX/usability work, instead of this only being a reactive quality-increasing activity.

The establishment of the field as a profession was accompanied by the standardization efforts conducted by ISO [18, 23]. Even though the role developed and an increasing number of people could claim this role from education and work [24, 25] usability professionals have always struggled with the justification of the role [26] and the call for a need to cost-justify its activities [27]. One approach of making the field more attractive was tighter integration with software engineering models [28, 29], but still integrating usability activities into agile development has challenges. In studies in practical settings, the main challenges found for the integration of usability activities were availability of IT professionals and time [30, 31]. The profession gained further popularity with the growth of accessibility [32] since legal binding obligations to adhere to accessibility requirements made it necessary to incorporate accessibility professionals. It did however not reach the level of being popular and one reason for this may be the view that the profession's work is too research focused, and does not include the demanded levels of agility.

10.2.3 The Integration of Usability Activities into Scrum in Practice

In Scrum the customer's needs and thereby the user needs are meant to be taken more into account than in older development processes, like Waterfall and RUP [2]. Through the product owner role, the priorities of the customer are considered. That does not necessarily mean that the users' needs are taken into account, because the customer representatives are not always the ones who will use the software. The Product Owner does not always state the needs for usability of the software in practice either [33]. One of the main conclusions in an extensive literature survey from 2010 on the integration of the usability needs into agile processes in the software industry is that the users´ needs were not sufficiently included in the agile development processes [34]. A recent literature study from 2015, states that usability methods are often used too late in the development of software in the industry and managing the product vision and time boxing the user-centred design work are well known continuous challenges [35].

Recent results show that many IT professionals frequently use some kind of usability techniques in Scrum projects, to support the four usability activities, and that they generally rate the techniques as being useful [5]. The highest rated usability techniques according to this study were workshops, informal usability evaluation with users and meetings with users. It is noticeable that all of these techniques are rather informal. Similar results have been presented in a study showing that all kinds of prototypes are used more frequently in Scrum projects than when using software development process made by the corresponding software company [36]. An extensive literature study shows that the most common usability techniques in agile development are low fidelity prototypes, user testing aimed at refining the prototypes in the next iteration and inspections [37].

Organisational and contextual settings for successfully integrating usability activities into agile software development have been explored in several research studies. Close collaboration between the development team and the usability professionals has been considered as one of the biggest success factors for integrating usability activities in Scrum projects [37, 38]. The usability professionals' understanding of their job role and the need to establish and communicate an overall team vision were pointed out as the two major themes highly important for the success of integrating user activities in agile development [39]. Often user experience issues are considered important both on strategic and operational level, but the current work processes and management styles can limit the impact of the usability professionals' work [40].

10.2.4 Diffusion and Fashion

Rogers [10] has shown that modelling diffusion processes typically result in a characteristic S-shaped curve. The curve traces the accumulated adoption of an innovation over time as it initially accelerates, reaches a peak rate of diffusion, and then decelerates as the level of saturation approach its maximum. Early on, when diffusion is slow, only those categorised as "innovators" and "early adopters" are interested in the innovation. The rate of diffusion increase as the interest of the majority of potential users peaks, influenced by factors such as the perceived maturity of the innovation, financial incentives provided by investors promoting the innovation, proximity and relationship to other innovation adopters, etc. Upon approaching saturation, the rate of diffusion decrease as the group categorised as "laggards" is the only group left in the population of agents that will eventually adopt the innovation.

The fashion metaphor has been widely used to describe and explain the popularity and rapid diffusion of certain "new" management ideas in the form of concepts, models, methods and techniques [41]. The metaphor draws on aesthetic fashions, notably the clothing industry, but according to Abrahamson [9] there are two important differences between management fashions and aesthetic fashions: “First, whereas aesthetic fashions need only appear beautiful and modern, fashionable management techniques must appear both rational (efficient means to important ends) and progressive (new as well as improved relative to older management techniques)” (p. 255).

Much research on the use of the fashion metaphor in management research builds on the perspective of new institutionalism [42–45] and describe how management ideas are chosen based on an organization’s wish to be more “fashionable” rather than actual needs. Czarniawska and Joerges [44] describe how organizations adapt to new ideas and how these ideas get their velocity, i.e. ability to spread widely, mainly from being in vogue in the sense that many people/organisations in different places and over time will “catch on” to fashionable ideas and use them. Fashion can thus explain “many puzzling developments in and between organizations” (p. 24). In other words the metaphor of fashion can be helpful in understanding changes that occur across an organisational field and even how institutions are altered or changed.

Clark [46] points out that organizational conformity to the latest fashion may result in disappointment and frustration as the models and techniques do not always live up to the expectations set for them. This in turn may result in the lack of institutionalization and organizations abandoning the ideas in favour of others in the process of becoming fashionable.

10.3 Method

The results and discussions in this chapter builds upon several qualitative interview studies with usability and user experience professionals and quantitative survey studies in agile projects. It is a synthesis chapter that integrates results on how IT professionals working with software development in agile projects conduct user-centred activities. An overview of the studies described as study 1–5 is given in Table 10.1. In total 37 IT professionals were interviewed and 74 took part in surveys.

Table 10.1 An overview of our studies that are integrated in this paper

Study no. (id)	Research method (year of study)	Information on the studies
Study 1 (S1)	Survey and interviews (2009)	25 IT professionals from 18 software companies in Iceland responded to the survey, 6 of the same respondents took part in interviews
Study 2 (S2)	Survey (2011)	49 IT professionals mainly in Sweden responded
Study 3 (S3)	Interviews (2010)	21 IT professionals from 14 companies in Sweden interested in usability and UX
Study 4 (S4)	Interviews (2013)	6 IT professionals in Sweden working at a software company developing a system for health care
Study 5 (S5)	Interviews (2013)	10 IT professionals in Iceland working at 10 companies with systems development

Hence, data was gathered from 111 professionals mainly in two countries. All the participants used agile processes and conducted some usability activities in their software development.

The motivation for Study 1 was to compare usability evaluation to other types of testing in Scrum projects. The participants were also asked about how they use Scrum in their projects and what the positive and negative aspects arise from using Scrum. The informants were mainly working at software companies developing various types of software, but some were working at banks and telecommunication companies developing software for that company. Around half of the companies had more than 40 employees.

The objective of Study 2 was to get an overview of what user-centred techniques were used by IT professionals working in Scrum projects mainly in Sweden. In this study the informants were also asked how they use Scrum and what the drawbacks and advantages are. One third of the respondents were working at companies having more than 250 employees, and one third had between 50 and 250 employees. Almost half of the respondents were working in Internet and e-commerce systems, and 20 % were working in the computer industry. The remaining respondents were working on telecommunication, in the financial sector, e-health or in other areas.

In Study 3 we interviewed IT professionals in four different roles: Team members, Scrum managers, business analysts and usability experts. The informants came from 14 companies in various organisational contexts. The main types of organisations were product development and consulting companies, or a combination of those. Some of the companies were international, having employees worldwide. The number of employees reached from 8 to 12,500. The purpose of the study was to understand the challenges that IT professionals interested in usability activities face when integrating usability activities in agile software development.

Study 4 was an interview study related to the introduction of a controversial eHealth system where Scrum was used by the vendor company. The purpose of the study was to evaluate the development process from a usability perspective. The interviewees were members of the Scrum team or managers. The vendor company had approximately 10,000 employees.

Lastly, study 5 was an interview study focusing on the strengths and weaknesses of using agile processes such as Scrum and Kanban. The interviewees were all managers or project managers are software companies in Iceland, developing various types of software. The sizes of the companies varied from 4 employees to over 500.

We have published seven papers describing and discussing the results of each of the studies [5–7, 47–50] and three theses have been written by HCI students [51–53]. Some material gathered in these studies has not been published in these scientific papers, but is used in this chapter.

We use a mixed methods approach in the synthesis research for this chapter. Data from five previously conducted surveys and interview studies are analysed again using the fashion metaphor described in Sect. 10.2.4, so all the material from these studies was analysed again for gathering new perspectives. The main themes that we analysed now were: reasons for using Scrum, consequences of using Scrum and usability activities in relation to Scrum.

The analysis used a mix of predefined categories as well as categories that emerged during the analysis. Two of the researchers analysed and coded the data together and discussed the results with the other authors. The quotes in the paper are from our informants. In some cases, we have adjusted the words used to make them more readable. We refer to the source of the particular result using study numbers in Table 10.1 (e.g., S1, S5).

10.4 Results

In the results section, we start by presenting results that indicate why Scrum can be seen as a fashion. We will elaborate on the reasons mentioned in S1–S5 for choosing Scrum as a systems development method. We will start by presenting results in relation to why companies choose Scrum, where the inspiration comes from and motivations behind the choice. We will also present results in relation to the consequences of choosing Scrum. We will also elaborate on perspectives found in S1–S5 on why conducting usability activities in Scrum have not become fashionable.

10.4.1 Reasons for Using Scrum

There are many reasons behind choosing Scrum as a development process. These reasons are often presented in a rational and objective manner, as if the choice was based on an objective and rational decision-making process. In our studies many say that the inspiration of using Scrum comes from the ideas inherent in the method, and these are very appealing to the system developers (S1, S2, S3, S4, S5). One of the inherent values that are appreciated in Scrum is the focus on "speed" and the "agile" way of working or as one of the informants describes it: "get the f***ing work done as fast as possible" (S4). One of the benefits of using Scrum, according to the interviewees, is that the team is to work independently, and this made the systems development model appealing to use (S3). By working independently, the team can focus on their task and does not have to use time on synchronizing with many others. The process leaves the decisions to the team, and this is inspiring for the system developers. Another appealing feature in Scrum is the focus on communication within the team, for example in the stand-up meetings (S3, S5). One can notice that these stand-up meetings are the most valued part of Scrum, and the most popular Scrum activity when implementing the process (S1).

Another advantage is the short sprints used, where a potentially shippable product is delivered to the customers after each sprint, which gives the developers a good pace to work in (S3, S4). One informant explains: "I guess for a programmer it's perfect to get a small piece of work that you can develop and deliver" (S3).

Additionally the team members mention that focusing on only several tasks during the sprint and not being interrupted all the time by requests from the customer is one of the advantages of Scrum. One informant explains that the team members are protected from interruptions from customers (S5). These sprints imply that the visions of the customers are considered through the whole development process. When following this process with short sprints one avoids discovering failure at the end of the whole systems development project when it is too late to do anything about the problems, as one of the informants described: “When we start to work and showing the end-users the product as often as we do using Scrum, we always get the feedback that now I understand what you were talking about. And then you can continue that discussion: Is this what we are showing you what you were expecting to get? And then you have this discussion all the way from the beginning instead of the ordinary way when you have 2 weeks to deliver when you are showing them: This is what you get and they go: Wooooaaaaa! We don’t want this. So we get all that very early in the project which means that when we have the final delivery we know that this is the product that the customer actually can use because it is well suited in their way of working and the kind of process that they need” (S3). The short sprints provided a suitable framework for managing the direction of the software being developed as the project’s vision could be described as a moving target depending significantly on external factors such as necessary integrations with third party systems, a precarious political situation regarding this kind of (nationally) unprecedented service, and rulings from national regulating authorities (S4).

The choice of Scrum as a development process is also based on previous development processes used and many informants position their choice of Scrum in relation to the waterfall model (see the citation above where the informant refers to “the ordinary way”). However, it seems to be easier to say what did not work in previous systems development processes (or waterfall), and why they did NOT work. It is more difficult to say what DID work with them, as this informant exemplifies: “I think the waterfall model is really bad.” (S3) Another informant describes the differences between Scrum and waterfall by saying: “But I think that maybe the most important thing with Scrum is the knowledge that it’s a fluid world. The requirements will change. And that’s in your mind set, so you’re more liable to cope with that. In a more traditional waterfall development you believe that these are the requirements and they’re going to remain fixed, which is never the case. So that’s probably the most important thing about agile methods that you realise that things are going to change as we go along” (S3). Moreover, as can be seen in the citation the choice of Scrum is motivated by the values inherent in Scrum regarding the inevitability of changing requirements. Many organisations tend to use their experience from their previous methods, choosing a development process that is different from the old process used. One informant describes the situation in Iceland by saying: “So my experience in Iceland is that usually companies have no strict processes. In many companies there is no process defined. A little bit of chaos or some kind of process but no formal process” (S5). Introducing Scrum to those companies gave the developers structure to the previous less organised way of working the informant explained (S5).

10.4.2 Consequences of Using Scrum

In our studies we have seen that most companies have adapted Scrum or introduced a few inspirations from Scrum, and they describe that they are using a "Scrumish method" (S3, S5). This can be seen as an indication that Scrum does not actually fit as a development process and one of the consequences of choosing Scrum was that it needed to be adapted.

Possibly some of the core values of Scrum are currently fashionable, such as speed and agility, and these are the ones organisations choose to implement. Most companies do however use the majority of the roles, activities and artefacts recommended in Scrum (S1, S2, S4, S5). Most Scrum projects have a Product owner, and a Scrum master but in some cases their roles have been omitted, replaced by a project manager or by members of the teams. Like one of our informants described: "We are not terribly concerned with the Scrum Master role. We have one (rotating) pair working outside the sprint to take care of technical support, disturbances and general application health not directly related to customer needs" (S2). The most popular activities used are sprint planning used for planning the scope of the starting sprint and stand-up meetings for checking the status of the sprint on daily basis. When looking at the artefacts, the least used artefact is the burndown chart (S1, S2).

Some informants find it problematic that the Scrum team is supposed to consist of team members that share the workload according to oral agreements made in the team. There is not really a good way to include specialist roles focusing on one of the quality aspects of systems development. Either the specialists are team members, or they are working outside the teams. None of these work situations are without problems, and organisations have tried out different solutions to solve this. Being outside can make the specialist feel like an "add on" to the project, and not a core member (S3), or as another expert explained it: "I worked in a small parenthesis outside of the team" (S4). In the interviews it became obvious that members of the development organisation were described and presented according to their background and education, as well as their speciality in systems development rather than as being team members. The specialist roles do not have an obvious place in the Scrum structure, and this is a problem for many projects (S4). The Scrum development process does not handle this need for specialists in the development projects. One of the usability specialists says: "I interfere in their process to help" (S3). Some of the specialists think that Scrum does not fit their needs, and that they would have liked a development process that supported their work in a better way.

Sometimes Scrum does not match with external requirements in the organisation (S3). Scrum does not fit since there are numerous other requirements due to adherence to standards, and the requirement of a specific area of application. Some projects need to dissolve the Scrum structure in order to comply with external circumstances, and the Scrum process does not really fit the circumstances of the organisation, as in this quote: "Right now we have mixed the teams up, because we have to do so at the end of our development cycle" (S3). One problem that was mentioned is that it is not possible to add functionality within a sprint, even though the customer really needs that functionality (S3). This makes Scrum somewhat inflexible despite the fact that it is presented as very flexible. Several organisations using Scrum feel that it is a problem that the sprints are fixed, and in some cases the Product owner has the rights to add emergency work during the sprint (i.e. even though this was not planned in the sprint planning). “Some requirements are urgent, cannot wait 2 weeks. We have created exceptions to the sprint process for specific tasks” (S5).

One disadvantage mentioned with Scrum is that there is no clear vision of the product, which can be problematic in several ways. One informant describes: “There is no clear description of what the system is going to be when the project starts, and Scrum requires that the procurer or customer is much more active all through the project. It is problematic for the procurer or customer not to have a clear vision of the product. They pay for something without having a clear idea of what they will receive.” (S3). Another informant describes the problems with the vision in this way: “But the problem is, that there is now no one that actually is responsible for putting this piece of functionality into the big picture. So there is no one responsible for the full user experience. That’s the problem. After a while you have added so many pieces that you don’t know where to put them anymore. And if you don’t have the vision clear in your head or on paper it’s starting to get quite difficult to know what to do with this piece of functionality and then you do something, just to squeeze it in. And that’s the reason for that I think it’s so important to do a thorough pre-study before prototyping and testing. Because if you have that it’s so much easier to prioritize and say okay say that from this vision we have decided to do this piece now and that piece then. At least we know where it all fits in. And then of course this vision will change all the time, because the market changes or whatever. But still you can work on the vision then and know where to put the pieces” (S3). Several other informants share this concern, that the holistic view for the system is missing, since the sprints slice the development project in small groups of functionality that become potentially shippable product after each sprint.

10.4.3 Usability Activities in Scrum Projects

Usability activities are mostly not an obvious part of Scrum projects, and usability experts face problems when integrating their work (S3, S4, S5). Activities such as usability evaluations, as for example defined by Benyon [54] are rarely used in the way described in such standard literature, despite the fact that the organisations had well educated usability experts in the teams [7].

Some organisations see usability activities as something completely different and less prioritized than the rest of the system development tasks and other user stories (S3). User stories connected to usability are placed on a separate Scrum board. The two Scrum boards live in parallel in the projects, and they had one Scrum board that was the “ordinary Scrum board” and one Scrum board that was the “usability

Scrum board". However, the most important thing in the project was to work with the ordinary Scrum board, and the user stories on the usability board were only dealt with if the system developers in the team had time. This was explained by one informant: "we had kind of a second Scrum board. Whenever there was time over, because often we could do it in half an hour to an hour. So then we could take one of these notes and fix it and just check, this is done, put it back and then he evaluated it. All this was kind of informal" (S3).

It is noticeable from the studies that some usability experts wrap usability problems in more fashionable concepts. One example is the usability expert who rephrases the usability problems as security problems, and makes them more interesting for the organisation in this way: "We are working on really enhancing the benefits of usability in the area of IT security" (S3). Another usability expert explains that the only way to make usability problems interesting in the organisation is to present them as new user stories that needs to be solved, not improvements to the old system from the usability perspective, or re-launching old user stories (S3).

Usability work is sometimes carried out as a completely separate activity from the work in the team (S3, S4), and sometimes due to the fact that the level of usability and the design needs attention (S4). The usability experts are not a part of the Scrum team, but merely give written input from a distance and are invited to sprint demos and project meetings, but otherwise they do not work closely with the development team. One informant explains her position in this way: "I have only been working with some kind of test of the system, and then given small comments on the user interface" (S4). The same informant also says that she was a part of the team during the most intense period of the development, as is described in this quote: "During the period that I worked full time with testing and evaluation, then I was participating in the stand-up meetings, but I did not really participate in the meetings" (S4). So, even though the informant was working actively on the project and technically included in its activities, she was not an active part of the team. Another usability expert who had been working at the company for 6 months, explains that he, when trying to find out what the team is doing, asks the Scrum team: "Can I be at your meetings?" (S3), i.e. from his perspective he is not included in the team. He explains that he is carefully taking small steps to change this culture.

One of the reasons frequently mentioned for why usability activities are not emphasised is that it is too time consuming (S1, S3, S4). When comparing why developers use some testing techniques less often than other techniques while developing their software, lack of time is the most frequent reason for not conducting usability testing (S1). This is also true, for security and performance testing. An interesting outcome from the comparison in the study (S1) is that the four core testing types, unit, integration, system and acceptance testing [55], are always budgeted for, but for the quality aspects like usability, performance and security are not in the budget by default and sometimes that is the reason for not conducting those (S1). Many informants mention the vagueness of the concept usability. It is often seen as something fuzzy, hard to measure and usability requirements are always changing (S1, S2, S3). This perspective could be one of the reasons for why the IT professionals see usability activities as time consuming.

One respondent explained that if the project is really agile, the changes are not that extensive each time and the importance of being quick to market is strong, so usability testing is really not needed, because a shippable product has been delivered and the customers can complain (S1, S3, S4). Furthermore, because the changes are small, extensive usability testing is not needed and is too expensive. "The main thing is to confess your fault and change quickly according to the customers' complaints so you can be very quick in adjusting to their needs" (S1) one of the respondents remarked. This respondent explained that asking for usability testing was really the customer's responsibility. We have also heard the opinion that IT professionals really want to gather feedback from users, but users were not always willing to take part because they were busy doing their own work and did not want to be involved in software development.

Some informants describe that they launch the system, and expect the users to tell if the system doesn't work while using the system (S1, S3, S5). One informant describes her experiences by saying: "As soon as the user gets the possibility to start clicking himself then it feels like their brain is finally working – they say: Oh now I understand even though you can simulate it on the whiteboard as well. But I think that is a big difference that we actually have a product in the end of every sprint." However, users seldom complain due to numerous reasons related to not having the vocabulary, power, and blaming problems with the system on themselves (S5). The system developers hence get very little feedback, and they interpret this silence as if the users are satisfied with the system.

10.4.4 Summary of Results

In our studies we have seen that Scrum has become popular for a number of reasons. In the following, we summarise the reasons and the consequences of using Scrum. Furthermore, we summarise the challenges of integrating usability activities in Scrum projects.

The main reasons we have seen of using Scrum are:

1. Many of our informants mention that the emphasis on speed, reflected in the short sprints and frequent deliveries, is very positive in Scrum.
2. The team members can focus on a few issues at a time and get the feeling of achievement every other week.
3. The team works independently and communication is encouraged within the team. This is described as beneficial for the work environment and group culture of the team.
4. The acceptance for requirements changing over time is seen as positive.

However, there's also a negative side of what is described as primarily positive. In our cases we have detected a number of aspects where Scrum falls short of being the ultimate process for all purposes for all actors involved. Below is a summary of the consequences of using Scrum:

1. Scrum does not seem to fit everyone, the respondents in our cases referred to themselves as using a "Scrumish method", indicating that they do not do Scrum by the book.
2. The consequence of the frequent deliveries in Scrum is that often there is no time for usability activities. We have seen that often the team members do not find time to gather feedback from users. This may be because of the emphasis on speed in Scrum and the changes of requirements from the users are not always welcomed by the team members. We have also seen that the developers rely on users to provide feedback after using the latest delivery for some time, but that is not often the case.
3. The team is supposed to provide expert knowledge but Scrum makes it hard for experts to fit into the team structure. The experts feel like outsiders in the process.
4. Due to the fragmented nature of the Scrum process it is hard for both the team and the customer to have a clear vision of the result of the process early on.

Additionally, we analysed the challenges for integrating usability activities in Scrum projects in practice. Our main results are summarised below:

1. The fragmented nature of Scrum, mentioned above, is fundamentally at odds with the methods of experts working with usability, as they want to grasp the whole picture from the beginning. Also this lack of a holistic approach leads to that none has the full responsibility of the whole user experience.
2. The usability experts do not become members of the teams, but give written feedback from a distance.
3. The usability aspects do not fit into the "quick" processes of Scrum and are thus perceived as too time consuming and costly and are often not even a part of the budget.

10.5 Discussion

In this section we discuss the two underlying questions in the paper: Why is Scrum so fashionable? and: How can usability activities be better integrated in agile projects?

10.5.1 Why Is Scrum so fashionable?

According to Abrahamsson [9] management fashion differs from aesthetic fashion in two ways. Firstly, fashionable management techniques need to come through as both rational and progressive and not only as beautiful and modern. Secondly, whereas aesthetic fashion is shaped by socio-psychological forces alone, management fashion is shaped by economical and technical forces ([9], p. 255). Many of the people we interviewed in our studies said that they were using a Scrumish method, only using bits and pieces from the Scrum process. One indicator of Scrum being a fashion, is the fact that it seems more important for many companies using Scrum to be able to say that Scrum is used, than it is to actually use the whole process and incorporate its ideas extensively. It is also obvious that many different actions are performed under the Scrum label. Scrum has become one of the most used methods for software development and this homogenization can be interpreted as an urge amongst organizations operating in the same field to show that they have knowledge of the latest management models [56]. This might thus indicate that the organizations using Scrum are interested in retaining legitimacy amongst their "peers" [42]. Scrum might thus be used as a tool in the employer branding of the companies so the employers are proud to be able to say they are using the fashionable Scrum process and use that as a sign of being one of the leading companies. However, it should be noted that few management models used for software development are fully incorporated without adaptations in an organisation. Also a model like Scrum can be seen as fashionable when it is chosen for its legitimizing effect rather than for practical reasons.

This does not mean that Scrum does not fit the purposes of these organizations. On the contrary, the process is put forth as a rational way to develop software as it allows for an iterative and swift development process, where the development team is able to focus on a few requirements at the time. Also, delivery through sprints provide continuous results which is seen as beneficial for the customer since they have control of the progression of the development. There are thus many rational reasons for using Scrum present in our data, which according to Abrahamsson [9] is necessary for a management technique to become fashionable. The question is rather if Scrum is always the best alternative, in every case where it is used. As listed above, in the summary of results section, there are several aspects of the Scrum process that are criticised by the IT professionals in the studied cases, indicating that other development processes might have been more useful but Scrum was chosen because it was fashionable. In a recent study, one of the conclusions is that IT professionals are leaning towards Kanban, since Scrum does not fit their needs [57].

From a diffusion perspective [10], the Scrum principles can be seen as a factor facilitating its diffusion. As mentioned above, the Scrum principles are few and flexible enough for organizations to be able to claim having adopted Scrum with minimal need for adapting their earlier way of working. In Rogers' terminology, Scrum offered a high *relative advantage* through its increasing legitimacy and fashionability while enjoying a high level of *compatibility* with previous processes, coupled with low *complexity* thanks to the comparatively lightweight principles. With Scrum being an intellectual innovation, not requiring any hardware or other specific artefacts to diffuse, its low complexity could also be seen as contributing to a higher level of *trialability* and *observability*. Thus, once considered fashionable enough, Scrum could diffuse rapidly as it did not require a significant investment in time or resources for an organization to climb aboard the Scrum bandwagon. Scrum could also reach a large portion of the software development community as the principles allowed for the process to be adopted, whether by name or in earnest, by a wide range of organizations. However, as there are other processes belonging to the agile family based on even fewer and more flexible principles that have not been widely adopted like the Kanban process, this could suggest that the perceived relative advantage of Scrum was the deciding factor. The successful diffusion of Scrum could then be attributed to an initial advantage over other Agile processes, such as XP, DSDM and Kanban, through serendipitous events, its popularity and legitimacy (*relative advantage*) then gaining faster than that of its competitors through a process of increasing returns as more and more organizations adopt Scrum.

10.5.2 How Can Usability Activities Be Better Integrated in Agile Projects?

One of the aspects that comes through in the accounts of the IT professionals in our cases is that Scrum is a development process that foremost suits the developers themselves. Usability experts and other experts find difficulties in becoming included in the development process as it is fragmented and the sprints provide only a short-term perspective. We have also seen that the responsibility for usability activities is not clearly defined. This is also enhanced by the lack of documentation in the Scrum process. Also it seems like the usability experts, at least in some cases, will refer to usability problems as for instance security problems to “sneak” the usability aspect into the process. This may also indicate that the usability aspects are out of fashion in the software development business and rebranding it is a way to make it more up to date and attractive [58]. There are of course many ways to better integrate usability activities in the different cases, adding a responsible person in some of the projects and better define the documentation in other projects, but there seems to be a lack of a holistic view on how the vision is for integrating user centred design in agile processes.

Management processes like Scrum may thus endure and be used successfully in the sense that they are perceived as useful by those using them, whilst they may fail in the sense that they do not create an adequate development process in terms of the final product or service from the end users’ point of view. Important factors may be overlooked in the process. This can be seen for example in one of our studies where the developers did not know how the customers use the developed product and the communication between developers and customers was not frequent. Even when the Scrum process is performed in business to business, the Scrum team may perceive it as problematic to involve the customer in the process and as illustrated in one of our studies, where we have seen that team-members are almost protected from customers. This shows that the urge to involve the user is low in that case. When the end user is not the same as the customer paying for the system the risk of the end users’ needs to be overseen in the development process is even greater. Agile processes strongly encourage customer participation, but what Scrum does not address is the user’s involvement. Often the difference between the customer and the user is not clear for agile software developers [57], so they think that by including the customer's perspective they have included the users' needs, which is not always the case in practice. Providing IT professionals with tools or techniques to plan user involvement, by defining who would be preferred participants, when it should preferably happen, where it should preferably happen and what should be the main goal of the user involvement could be one way of better integrating usability activities in agile projects.

Some software development companies market themselves as producing software with good user experience, e.g. Google. However, surprisingly few have adopted usability methods and successfully incorporated these into development practices. This can for example be seen in the results in one of our interview studies, where only very few interviewees named particular usability methods such as heuristic evaluation and user testing. In the same study, the interviewees contacted users informally and usability activities were not explicitly integrated into the development process. However, developers in general believe that they already develop highly usable software. Even if it leads to (even) better software from the user's perspective, it is not obvious to a software development company why they should extend usability practices beyond what they are already doing. This is a significant difference from the greening trend [41] where many consultants relied on rhetoric and associating environmental responsibility with established traits and values, e.g. by calling it a challenge that only the flexible and far-sighted companies would prosper from. Fineman [41] describes how the greening trend of incorporating environmental responsibility into management methodologies in the 1980s and 1990s was hard to sell on its altruistic values alone, they also had to satisfy more salient management issues such as profitability, efficiency, and performance. Growing public concern and environmentalist movements were not enough. Though it has become fashionable to appear green, Fineman [41] concludes that actually embracing all of the diverse environmental issues applicable to a corporation remains a quite complex and confusing task for managers.

There are many similarities in these two trends, the greening trend and the user experience trend. It has been hard to sell to managers that usability is important at least in the 1980s and 1990s and the reasons were similar, the managers emphasise profitability, efficiency, and performance that are hard to associate with usability work. Nowadays, UX has become more fashionable in the software industry and it is commonly respected by IT professionals that UX activities are important.

Still, integrating usability and user experience activities in Scrum projects remains quite a complex and confusing task and it remains a challenge for HCI researchers to help them to solve that task. Since usability professionals want to grasp the whole picture of the user experience from the beginning of the software development, the fragmented nature of Scrum is fundamentally at odds with their vision of how to approach their work. The usability aspects do not fit into the "quick" processes of Scrum and are thus perceived as too time consuming and costly and are often not even a part of the budget nor are the usability professionals a part of the teams, and end up being outsiders. Also this lack of a holistic approach means that none has the full responsibility for the entire user experience. The understanding of the reasons for choosing Scrum, the challenges for using the process and the challenges for integrating usability activities into Scrum explained in this paper, gives a good basis for both describing new ways for supporting usability professionals and for planning research on this topic.

10.6 Conclusion

The forces behind the development of the Scrum process were probably the rapidly increasing demand for new IT systems in the late 1990s, creating a need for new methods and techniques for managing the software development process. One of the approaches is the iterative way of developing software, which is illustrated by the sprints in Scrum. However, the downside can be that the holistic view is missing in the development process and that it may be difficult for the customers to continuously receive and implement new features of the system and learn how to use those.

In some cases, a certain development process, selected for being fashionable, may be used to produce a solution aimed at improving the work environment for a particular group. Management models that are chosen for the wrong reasons may endure and be used successfully in the sense that they are perceived as useful processes by those using them, in the case of this paper by IT professionals, whilst they may fail in the sense that they do not enable other stakeholders to improve their users’ work environment or create an adequate IT system from the users´ point of view. The consequence of Scrum being fashionable is that important factors may be overlooked in the development process of the IT system. The bottom line is that management processes that seem to work for IT professionals may not give a final result that works for those who will actually use the IT system. It remains a challenge for us HCI researchers to help IT professionals to integrate the users’ perspectives in their agile software development.

Acknowledgement We would like to thank COST Action IC0904 Twintide for the financial support that they have provided for completing study 3. Also, we would like to thank Professor Emeritus Liam Bannon for his valuable feedback on a draft of this paper.

References

1. Kruchten P (2004) The rational unified process: an introduction. Addison-Wesley Professional, Boston
2. Schwaber K (1995) Scrum development process. In: Business object design and implementation. Springer, London, pp 117–134
3. Beck K, Beedle M, Van Bennekum A, Cockburn A, Cunningham W, Fowler M, Thomas D (2001) Agile manifesto

4. VersionOne (2014) State of Agile Survey. Online at: http://www.versionone.com/pdf/stateof-agile-development-survey-ninth.pdf
5. Jia Y, Larusdottir MK, Cajander Å (2012) The usage of usability techniques in Scrum projects. In: Proceedings of the HCSE conference, Toulouse
6. Cajander Å, Larusdottir MK, Gulliksen J (2013) Existing but not explicit – the user perspective in Scrum projects in practice. In: proceedings of INTERACT conference
7. Larusdottir M, Cajander Å, Gulliksen J (2014) Informal feedback rather than performance measurements–user-centred evaluation in Scrum projects. Behav Inform Technol 33(11):1118–1135
8. Baxter G, Sommerville I (2011) Socio-technical systems: from design methods to systems engineering. Interact Comput 23(1):4–17
9. Abrahamsson E (1996) Management fashion. Acad Manag Rev 21(1):254–285
10. Rogers EM (2003) Diffusion of innovations, 5th edn. Free Press, New York
11. Sutherland J (1995) Business object design and implementation workshop. In: Addendum to the proceedings of the OOPSLA 1995 on object oriented programing system, languages and applications. Austin, Texas, USA
12. Takeuchi H, Nonaka I (1986) The new new product development game. Harv Bus Rev 64(1):137–146
13. Curtis B, Kellner MI, Over J (1992) Process modeling. Commun ACM 35(9):75–90
14. Boehm B (2006) A view of 20th and 21st century software engineering. In: Proceedings of the 28th international conference on software engineering. ACM Press, Shanghai, China
15. Basili VR, Turner AJ (1975) Iterative enhancement: a practical technique for software development. IEEE Trans Softw Eng SE-1(4):390–396
16. Schwaber K, Beedle M (2002) Software development with Scrum. Prentice Hall, Upper Saddle River
17. Card SK, Newell A, Moran TP (1983) The psychology of human-computer interaction. L. Erlbaum Associates, Hillsdale
18. International organisation for standardisation: ISO 9241-210:2010 (2010) Ergonomics of human-system interaction – Part 210: Human-centred design process for interactive systems. International Organisation for Standardization, Geneva
19. Hassenzahl M, Law ELC, Hvannberg ET (2006) User experience-towards a unified view. UX WS NordiCHI 6:1–3
20. Law ELC, Roto V, Hassenzahl M, Vermeeren AP, Kort J (2009) Understanding, scoping and defining user experience: a survey approach. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, New York, pp 719–728
21. Wale-Kolade AY (2014) Integrating usability work into a large inter-organisational agile development project: tactics developed by usability designers. J Syst Softw 100:54–66
22. Plonka L, Sharp H, Gregory P, Taylor K (2014) UX design in agile: a DSDM case study. In: Agile processes in software engineering and extreme programming. Springer, Cham, pp 1–15
23. International organisation for standardisation: ISO 9241-11: 1998 (1998) Ergonomic requirements for office work with visual display terminals. International Organisation for Standardization, Geneva
24. Gulliksen J, Boivie I, Göransson B (2006) Usability professionals—current practices and future development. Interact Comput 18(4):568–600
25. Boivie I, Gulliksen J, Göransson B (2006) The lonesome cowboy: a study of the usability designer role in systems development. Interact Comput 18(4):601–634
26. Gulliksen J, Lantz A, Boivie I (1999) User centered design in practice-problems and possibilities. Royal Institute of Technology, Sweden
27. Mayhew DJ, Bias RG (eds) (1994) Cost-justifying usability. Academic, Boston
28. Seffah A, Gulliksen J, Desmarais MC (2005) An introduction to human-centered software engineering. In: Human-centered software engineering—integrating usability in the software development lifecycle. Springer, Dordrecht, pp 3–14

29. Göransson B, Lif M, Gulliksen J (2003) Usability Design-extending rational unified process with a new discipline. In: Interactive systems. Design, specification, and verification. Springer, Berlin/Heidelberg, pp 316–330
30. Bruno V, Dick M (2007) Making usability work in industry: an Australian practitioner perspective. In: Proceedings of the 19th Australasian conference on computer human interaction: entertaining user interfaces. ACM, New York, pp 261–264
31. Cajander Å, Nauwerck G, Lind T (2014) Things take time: establishing usability work in a university context. In: Proceedings of the EUNIS 2014: higher education in the digital era. ICT Services and System Development, Umeå University
32. Persson H, Åhman H, Yngling AA, Gulliksen J (2014) Universal design, inclusive design, accessible design, design for all: different concepts—one goal? On the concept of accessibility—historical, methodological and philosophical aspects. Universal Access in the Information Society, May 2014, Springer, pp 1–22
33. Singh M (2008) U-SCRUM: an agile methodology for promoting usability. In: Agile, 2008. AGILE'08. Conference. IEEE, pp 555–560
34. Sohaib O, Khan K (2010) Integrating usability engineering and agile software development: A literature review. In: International Conference on Computer Design and Applications (ICCDA), 2010, vol 2. IEEE, pp V2–V32
35. Brhel M, Meth H, Maedche A, Werder K (2015) Exploring principles of user-centered agile software development: a literature review. Inf Softw Technol 61:163–181
36. Larusdottir MK, Haraldsdottir O, Mikkelsen B (2009) User involvement in Icelandic Software Industry. In: Proceedings for the 2nd international workshop on the interplay between usability evaluation and software development – I-Used at the INTERACT conference 2009. Uppsala, Sweden, pp 51–52
37. Silva da Silva TS, Martin A, Maurer F, Silveira MS (2011) User-centered design and agile methods: a systematic review. In: AGILE, pp 77–86
38. Kuusinen K, Mikkonen T, Pakarinen S (2012) Agile user experience development in large software organization: good expertise but limited impact. Human-Centred Software Engineering, Toulouse
39. Kollmann J, Sharp Hl, Blandford A (2009) The importance of identity and vision to user experience designers on agile projects. In: Proceedings of the Agile 2009 conference, pp 11–18
40. Kuusinen K, Väänänen-Vainio-Mattila K (2012) How to make agile UX work more efficient: management and sales perspectives. In: Proceedings of NordiCHI 2012 coference. ACMPress, Copenhagen, Denmark, pp 139–148
41. Fineman S (2001) Fashioning the environment. Organization 8(1):17–31
42. Mayer JW, Rowan B (1977) Institutionalized organizations – formal structure as myth and ceremony. Am J Sociol 83(2):340–363
43. DiMaggio PJ, Powell WW (1991) Introduction. In: Powell WW, DiMaggio PJ (eds) The new institutionalism in organizational analysis. The University of Chicago Press, Chicago
44. Czarniawska B, Joerges B (1996) Travels and ideas. In: Czarniawska B, Sevón G (eds) Translating organisational change. de Gruyter, Berlin
45. Czarniawska B, Sevón G (2005) Global Ideas: how ideas, objects and practices travel in the global economy. Liber, Malmö
46. Clark T (2004) The fashion of management fashion: a surge too far? Organization 11(2):297–306. doi:10.1177/1350508404030659
47. Larusdottir MK, Bjarnadottir E, Gulliksen J (2010) The focus on usability in testing practices in industry. Paper in the human computer interaction symposium at the World Computer Congress
48. Larusdottir MK, Cajander A, Gulliksen J (2012) The big picture of UX is missing in Scrum projects. In: Proceedings of the International Workshop on the Interplay between User Experience and Software Development – at NordiCHI
49. Scandurra I, Holgersson J, Lind T, Myreteg G (2013) Development of novel eHealth services for citizen use: current system engineering vs. best practice in HCI. In: Proceedings of INTERACT conference

50. Larusdottir MK, Cajander A, Simader M (2014) Continuous improvement in agile development practice – the case of value and non-value adding activities. In: Proceedings of the HCSE conference 2014
51. Bjarnadóttir E (2009) Analysis of software testing in Icelandic Scrum projects. BS thesis, Reykjavik University
52. Jia Y (2012) Survey on Scrum and UCD. Master thesis, Uppsala University
53. Simader M (2013) Insights into waste in agile software development. Master thesis, Reykjavik University
54. Benyon D (2013) Designing interactive systems: a comprehensive guide to HCI, UX and interaction design. Pearson Education, London
55. Graham D, Veenedaal EV, Evans I, Black R (2007) Foundation of software testing: ISTQB certification. Thomson, Belmont
56. Sahlin-Andersson K, Engwall L (2002) Carriers, flows and sources of management knowledge. In: Sahlin-Andersson K, Engwall L (eds) The expansion of management knowledge – carriers, flows and sources. Stanford University Press, Redwood City
57. Law ELC, Lárusdóttir MK (2015) Whose experience do we care about? Analysis of the fitness of Scrum and Kanban to User Experience (UX). Int J Hum Comput Interact 31(9):584–602. doi:10.1080/10447318.2015.1065693
58. Erlingsdóttir G, Lindberg K (2005) Isomorphism, isopraxism and isonymism: complementary or competing processes? In: Czarniawska B, Sevón G (eds) Global Ideas: how ideas, objects and practices travel in the global economy. Liber, Malmö, pp 47–70