



Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A license to kill – Improving UCSD in Agile development

Marta Larusdottir^{a,1}, Jan Gulliksen^{b,*}, Åsa Cajander^{c,2}^a Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland^b KTH Royal Institute of Technology, Lindstedtsv. 3, floor 4, 100 44 Stockholm, Sweden^c Uppsala University, ITC house 19, Box 337, 751 05 Uppsala, Sweden

ARTICLE INFO

Article history:

Received 5 January 2015

Revised 16 January 2016

Accepted 21 January 2016

Available online xxx

Keywords:

User-centered system design

Agile development

Scrum

ABSTRACT

Agile development processes, such as Scrum, focus on communication, developer collaboration and delivery of working software early and continuously. User-centered systems design (UCSD) is a process emphasizing usability and the user experience throughout the system life cycle. It highlights the UCSD activities: understanding the context of use, iterative prototyping to explore the design space and active collaboration with users throughout the software development. Agile processes are by many assumed to address similar issues as UCSD, hence, by applying Agile processes the systems would become usable for the end-users and their user experience should improve. This paper discusses and interprets findings on UCSD activities in Agile projects in practice, that are analyzed according to the fundamental principles from the Agile manifesto. We show that Agile development has much to gain from integrating UCSD, and give guidance on how to integrate UCSD in Agile processes. User experience (UX) professionals need a more explicit role in the Agile projects and more authority. We recommend that they receive a “license to kill” design suggestions that are not perceived as useful by the users.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Software development used to be conducted using a stepwise structured process with clearly specified activities and deliverables (Sommerville, 1996). Particularly the waterfall model was an important early software engineering processes, as it became more possible to plan and predict the work (Salah et al., 2014) long before anybody considered usability or the user experience (UX). Kirsten Nygaard was one of the early pioneers to introduce the human factor in the software development process, showing that programming is a social activity that requires specific attention to increase its efficiency and effectiveness (Nygaard, 1990). In the 1990s the concept of an iterative software development processes appeared, particularly with the purpose of minimizing risk (Boehm, 2003). Iterative design is also one of the cornerstones of user-centered systems design (UCSD), even though the User

Experience (UX) professionals' community¹ has a slightly different meaning to the concept of iteration.

In the transition from the traditional waterfall model of development towards Agile Olsson et al. (2012) have shown how “de facto” standards for software development support this transition. They show evolving software development practices moving from Agile towards continuous deployment of software and a future in which R&D works as an “innovation experiment system”. Today Agile development methodologies, particularly Scrum, have become more or less a de facto standard for software development. Many of those arguing for Agile methodologies claim that you automatically will develop usable systems and in a user-centered fashion by using these methodologies, and one of those who have argued in writing is Baxter and Sommerville (2011). The question is whether there is any truth behind such a claim?

We have over the years conducted numerous interview and survey studies about this topic and gathered substantial experiences through field studies and action research projects. The purpose of this paper is to bring together ours and others published studies

* Corresponding author. Tel.: +46 730 373930.

E-mail addresses: marta@ru.is (M. Larusdottir), gulliksen@kth.se (J. Gulliksen), asa.cajander@it.uu.se (Å. Cajander).¹ Tel.: +354 599 6200.² Tel.: +46 70 425 0786.¹ We refer to the professionals working with UCSD activities as UX professionals, regardless of if they have this title or not. The international association for people working with UX and UCSD recently changed its name from Usability Professionals Association (UPS) to UX professionals Association (UxPA).

together with additional experiences to make some general conclusions about Agile and UCSD. We will investigate and compare Agile development and UCSD in theory and practice. Our starting point is the belief that practice informs theory and theory informs practice synergistically. Even though they provide two separate perspectives, they influence each other. Agile development in theory may not be the same thing as Agile development in practice, but there are indeed common ideas and values. Practice and theory change over time, and in this change process they influence each other.

Throughout our research we have increasingly come to the understanding that a high degree of usability is a quality of immense importance for interactive software and that Agile approaches, despite their increasing popularity, were not in themselves able to guarantee to deliver this quality. Therefore the justification for our research is to realize the potential of all of our previous research in being able to communicate a few lessons learned, and thus that we based on the findings could be able to generalize the findings and deliver a few lessons learned for the benefit of others.

2. Background

Following we will outline the theoretical background behind UCSD on one side and on Agile software development processes on the other hand. The purpose is to show that although similar in scope there are fundamental differences in attitudes and values in relation to the processes.

2.1. Process perspective in User-Centered Systems Design

There is a large variety in the different types of user-centered design (Iivari and Iivari, 2006). Therefore we want to discuss it in relation to a stronger concept in which the word “systems” has been added to emphasize that we not only develop a new user interface, but an entire socio-technical system. We adhere to a definition of User-Centered Systems Design (UCSD) as a process focusing on usability throughout the entire development and further throughout the system life cycle (Gulliksen et al., 2003). It is a software development process focused on delivering the highest possible level of usability. Relating to usability as: “The extent to which a product, system or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.” (ISO, 2010).

From a process perspective UCSD can be understood in the following way: developing usable systems is about understanding the users and the context of use (usually through field studies and observations), turning the requirements of this process into designs (of various degrees of fidelity) and evaluating these design with users in context, all according to a predefined iterative and incremental participatory process.

An important process aspect of UCSD is its contextual nature. Field studies, observations and evaluations should be done in context, but also the design and development has plenty to benefit by being done in context with close proximity and access to the users. Additionally, it is important to understand the context of the development project as such. The *Context of Design* is “the socio-technical system comprising the design and development process in a lifecycle perspective” (Svanæs and Gulliksen, 2008). According to the definition a number of different factors affect the ability to develop usability by considering the context of design. Following is a non-exhaustive list: “The organizations involved, their relations and agendas, internal factors in the developer organizations, software development methodology and tools, maturity levels, internal factors in the client organizations, customer-developer legal relationships (e.g. contracts, tender), handover issues,

organizational stability, lifecycle perspective, and conflicting requirements” (Svanæs and Gulliksen, 2008). Therefore the lifecycle perspective of UCSD is required to bring in all values and assumptions into the process.

Understanding the current context of use to be able to specify the future context of use is a demanding task involving substantial amounts of design (Hardenborg, 2007). Creative design methods that involve users as active participants and engages developers to directly interact with users is important. It is an excellent learning opportunity for all parties involved, in line with cooperative design (Greenbaum and Kyng, 1991). Prototyping and design also play an important role for brainstorming new solutions with users. Conceptual design ideas are tried out with prototypes, the interaction is developed and detailed graphical and industrial design aspects are analyzed to gather further requirements (Royce, 1970). Prototypes are excellent for explaining the future use situations for non-computer scientists and for negotiating different design solutions and to gather feedback from the stakeholders (Sefelin et al., 2003). They serve an important purpose for all parties involved and can be used for games and role-play, for formative evaluation and for education and learning.

The definition of UCSD is based on the ISO 9241-210 Human centered design for interactive systems (ISO, 2010). The standard serves as guidance to project managers on how to conduct a user-centered project and hence focus on design in relation to the software engineering process. UCSD nowadays relies heavily on active involvement of usability professionals; a role that has been defined and further specialized over the last decades (Boivie et al., 2006; Gulliksen et al., 2006; Gulliksen et al., 2004). Recently in practice, the UX team has replaced the role of the “lonesome usability cowboy” (Sy, 2007; Sy and Miller, 2008).

2.2. Scrum – the most common Agile development process

Scrum is one of the most common Agile, lightweight processes that can be used to manage and control software and product development using iterative, incremental practices. According to the inventors Scrum is the fastest, easiest to implement, and most widely accepted Agile process (Schwaber, 2009; Schwaber and Beedle, 2002). It provides a mechanism to improve the communication between developers and clients and to take the focus away from the programming, putting additional emphasis on the planning and goal setting of program development.

The paper initiating Scrum had the title “The new new product development game” (Takeuchi and Nonaka, 2016). 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.” [ibid].

A characteristic of Scrum is the observation that small, cross-functional teams historically produce the best results. Scrum is based on a rugby metaphor in which the team’s contribution is more important than each individual contribution.

Scrum teams typically consists of people with three major roles:

- a Scrum Master that acts as project manager/buffer to the outside world
- a Product Owner that represents stakeholders, and
- a Team of developers (less than 10).

Some of the more important artifacts and ceremonies within Scrum is the Sprint, which defines 15–30 days iteration, the “Product backlog” of requirements (managed by the Product Owner), decided at the Daily Scrum (which is the daily meeting that is required).

Scrum allows the flexibility to work with UCSD activities during the project, but there is no or little explicit support for a specific UX role or for setting and meeting quality goals. Cockburn and Highsmith (2001) argued already in 2001 for a role to safeguard the user perspective in projects, but this is very different from the classic UX role. There is little low fidelity prototyping in the Agile processes in general and in Scrum in particular. Agile processes do not explicitly state that prototypes should be made and shown to users, but best practices may be added to extend the emphasis on, for example, UCSD activities. Often usability people involved in Scrum projects are advised to do the UCSD activities early, before the actual development projects start in, what is referred to as “the Magical Sprint Zero” (Najafi and Toyoshiba, 2008).

Some organizations even argue that such an approach should be systematized (Jakobsen and Johnson, 2008). Many of the usability professionals we have interviewed claims that if design shall happen in the project, it needs to take place before the start of the actual Scrum project. In some cases this approach is referred to as BDUF (Big Design Up Front) (Boehm, 2003).

3. Basic values in Agile development and UCSD

UCSD and Agile fundamentally have different sets of basic values. In the Agile manifesto (Beck et al., 2001) the core values of Agile development was agreed upon. The basic values of UCSD has been outlined in the definition and key principles of UCSD (Gulliksen et al., 2003). In this context we define basic values as ways of thinking and a basis for interpretations. IT professionals are continuously exposed to various situations and impressions, and their ability to simplify, create order and interpret these is crucial for their survival. Basic values are needed to create order as a starting point of further interpretation. The creation and reproduction of their basic values will inevitably affect the way of acting and thinking. Basic values make us see things differently and the perspective used determines many essential characteristics of our actions (Cajander et al., 2008; Nurminen, 1987).

The discussion of basic values in systems development is not new in HCI, and the conflict between different values in systems development has been extensively discussed, see for instance (Boivie, 2005; Greenbaum and Kyng, 1991; Kammersgaard, 1990; Orlikowski and Gash, 1994; Persson, 2004). Bannon concluded: “Part of the problem resides in an implicit view of ordinary people which, if surfaced, would seem to treat people as, at worst, idiots who must be shielded from the machine, or as, at best, simply sets of elementary processes or “factors” that can be studied in isolation in a laboratory.” (Bannon, 1995). In software engineering the systems theoretical perspective, (Nurminen, 1987) and the systems perspective (Kammersgaard, 1990) is deeply rooted in software engineering projects. Many approaches to systems development originate in an engineering oriented worldview, closely related to the systems theoretical perspective. This basic value is conflicting with the basic value of UCSD.

3.1. Related work on the integration of Agile and UCSD

Many researchers have analyzed and tried to understand the conditions under which Agile and UCSD may work together. Following we will present related research on the integration of Agile and UCSD.

Chamberlain et al. (2006) conducted a field study to investigate the integration of UCSD and Agile. They conclude that a successful integration requires balancing between each of the disciplines in the team and that sufficient resources for the work are provided. Additionally all key members must be involved in key decision points in the projects and that users play an important

part in the project. Blomkvist (2006) claims that Agile processes do not inherently provide the required support for UCSD. As an example, iterative development is fundamental to both UCSD and Agile development but the views on and definition of iterative is substantially different (Blomkvist, 2006). Nevertheless, the basic values and specific methods of Agile development may have the potential to work well together with UCSD. Ferreira et al. (2011) conducted an observational study of a mature Scrum team in a large organization, and their interactions with the UX professionals working on the same project. They concluded that the cooperation between the Agile developers and UX professionals was achieved through on-going articulation work by the developers, who were compelled to engage a culturally distinct UX design division. Constantine has also reworked his Usage-centered design methodology (Constantine and Lockwood, 2003) to become more lightweight (Constantine, 2002). He claims that his model-based approach focuses on usability and user interface design, and even more so when it has become turned into an Agile version of the same overall methodology. The user role and the basic principles of UCSD seem very distant, although usability, as a quality criterion, is considered. The UX professional role is also very important for managing the user perspective in the project. McInerney and Maurer (2005) interviewed three usability specialists in Agile projects. They were all very positive to their ability to manage UCSD activities in the Agile projects, and although they could not prove any improved effects in the resulting projects, they were positive to their ability to contribute, and did not see any negative effect from the fact that the projects followed an Agile approach (McInerney and Maurer, 2005).

Close collaboration between the development team and the UX professional has been considered as one of the biggest success factors for integrating UCSD activities in Scrum projects (Da Silva et al., 2011; August, Kuusinen and Väänänen-Vainio-Mattila, 2012). The usability professionals’ understanding of their job role and the need to establish and communicate an overall team vision was pointed out as the two major themes highly important for the success of integrating user activities in Agile development (Kollmann et al., 2009). 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 professional’s work (Kuusinen and Väänänen-Vainio-Mattila, 2012). One of five principles suggested in a recent literature study on the integration of UCSD activities in Agile development is continuous stakeholder involvement (Brhel et al., 2015). The authors claim that there is a lack of empirical studies systematically investigating the stakeholder involvement. Balancing the amount of upfront work and synchronizing between UX designers and software developers (chunking design to fit Agile sprints) have been found to be one of the main challenges in Agile user-centred design integration (Brhel et al., 2015; Rosson and Carroll, 2001)

Agile approaches emphasize the need to speak directly with the customer, but defining who the customer is can be difficult and not at all easy for the Agile professionals (Law and Lárusdóttir, 2015), some professionals refer to the product owner as the customer, some to the person paying for the software and some to the actual users (Law and Lárusdóttir, 2015). Following the UCSD tradition more than 70% of Agile professional did gather feedback from users on their software being developed; around half of them did gather feedback from client and only around 25% from customers (Larusdottir et al., 2015).

The overall impression from the related work is that it is indeed possible to conduct user-centered work in Agile projects, but that the success of it mainly is based on personal matters and relationships and that the Agile processes in itself does not provide sufficient support for UCSD.

Table 1

An overview of the studies that are analyzed in this paper.

	Research method	No. of participants
Study 1 (S1)	Survey	82 IT professionals in country 1 Published paper: Larusdottir et al., 2009
Study 2 (S2)	Survey and interviews	25 IT professionals from 18 software companies working on Scrum projects in country 1 in the survey, 6 IT professionals in interviews Published paper: Larusdottir et al., 2010
Study 3 (S3)	Survey	49 IT professionals working in Scrum projects mainly in country 2 Published paper: Jia et al., 2012
Study 4 (S4)	Interviews	21 IT professionals interested in usability and UX in country 2 3 Published papers: Cajander et al., 2013 , Larusdottir et al., 2013 , Larusdottir et al., 2012
Study 5 (S5)	Interviews	10 IT professionals in country 1 Published paper: Larusdottir et al., 2014

4. Method

In this section we first introduce the idea of the study in this paper, then we explain the synthesis process and the studies analyzed.

4.1. The idea of the study

This paper builds upon qualitative interview studies and surveys with usability and user experience professionals in Agile projects. The purpose of the study in the paper was to understand in what way usability and user experience professionals are able to apply UCSD activities in their projects and what the problems and possibilities for improvements are. The underlying question in the paper is: how well do the core Agile values fit for the usability and user experience professionals' activities? It draws upon discussions and analysis with practitioners and peer researchers on the integration of UCSD work in Agile projects.

4.2. The synthesis process

This is a synthesis paper that integrates results from several interview and survey studies related to the research topic. The authors analyzed the interview and survey results and the overall conclusion derived by clustering and relating our findings to the four main principles of the Agile manifesto. Additionally, we interpret the results according to our previous experiences and previous research conducted by others and us.

In total 37 IT professionals were interviewed and data was gathered from 158 professionals in the two countries using Agile processes. The studies listed in [Table 1](#) focus on the defined subjects that were analyzed and interpreted. This paper, however, summarizes our interpretations from these 5 studies and gathers our understanding and experiences on how the user perspective could be integrated in Agile software development on a higher and more aggregated level. We have analyzed our interpretations according to the four main principles of the Agile manifesto ([Beck et al., 2001](#)), and we refer to the source of the particular result by the numbers of the studies in [Table 1](#), S1–S5. Finally our overall conclusions on the problems and possibilities of Agile and UCSD have been discussed in several research seminars and workshops with fellow researchers and industrial collaborators, to help validate the findings and conclusions.

4.3. The studies analyzed in this paper

The study presented in this paper is a synthesis of the following studies: survey studies on how IT professionals working with software development conduct user-centered activities in the two countries are described as study 1, 2 and 3 in [Table 1](#). The motive of study 1 was to gain an overview of what user-centered techniques were used by IT professionals in one of the countries. Study 2 focused on comparing usability evaluation to other types of testing in Scrum projects and study 3 focused on gaining an overview of what user-centered techniques are used by IT professionals in Scrum projects mainly in the other country. In that study, interviews were also conducted. An overview of the background papers describing the studies is presented in [Table 1](#).

Additionally two interview studies are described in study 4 and 5. An extensive interview study was conducted with IT professionals interested in usability and UX issues, performing their work in Scrum projects named study 4. The goal of that study was to extend the understanding of the challenges that IT professionals have while integrating user-centered activities in Scrum projects. This study has been presented in three published papers, analyzing the results related to user-centered evaluation, user experience evaluation and the user perspective in Scrum Projects. In study 5, the value adding and non-value adding activities in Agile development was analyzed.

5. Empirical results

Based on the previous studies we can conclude that IT professionals working in Scrum projects are greatly affected by the constraints that the Scrum process imposes. We have analyzed our findings according to the four values in the Agile manifesto ([Beck et al., 2001](#)). Particularly we have used the 12 principles behind the manifesto to structure our findings.

5.1. Individuals and communication

The Agile manifesto ([Beck et al., 2001](#)) describes: "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."

Our results show that emphasizing the importance of considering and engaging individuals is similarly important in Agile and UCSD (S4, S5). Agile, however, do not primarily think about the users when they focus on individuals as is done in UCSD (S5). Our research shows that it is often unclear who is responsible for the interaction between the team and the users (S4).

Also our research shows that UX professionals rarely fit in the team culture (S4). From the results we can see that highly motivated individuals for integrating UCSD work in Scrum do not always have the environment and support they need to be able to do that work (S4). The team culture is strong in Scrum, and consequently the UX professionals do not feel that they belong to the team, they find it hard to fit in and to be able to do their job as they are motivated to do UCSD activities. One UX professional stated: "I'm like an ADD on", when explaining his collaboration with the team members (S4).

The autonomy of the development team is strongly promoted in Agile methodologies and trusting the team's ability to distribute and deliver on tasks by themselves is seen as an important asset in the development work (S2, S3, S4, S5). As UX professionals are not by default a part of the culture, this autonomy makes it somewhat difficult to work with UCSD in the Agile setting.

Trusting the team to do the job is a good principle, but often the team members and UX professional do not agree what it actually mean "to get the job done" (S4). One UX professional explained that developers often think they are done with a user

story, but the UX professional finds usability problems and has to ask the developers to make changes to the software (S4). In one company they extended Scrum with an extra definition of done to solve this problem. When the software developers are done, they hand it over to the UCSD experts who work with the user story until it is “done done”.

Moreover, when inquiring about the responsibility for usability, the answers were manifold (S4, S5). This shows that IT professionals interested in including usability activities in their daily work do not have a common understanding of who is responsible for this important quality criteria in software development. This could make it harder for the IT professionals to fight for the usability of the product. IT professionals also have different views on how the emphasis on usability could be integrated and extended. Some say that the product owner should be asked to take more responsibility about usability while others think that the team members should be trusted to do more usability work (S4).

The development of methods and processes to support development work and user experience activities is of course one way of increasing the quality of software development but it can never replace the need of great talents in the work. An exceptional designer may be able to complete wonders even if there are big deficits in the development methods and processes, simply by using the experience and talent. Talent may be a project's biggest asset, but with great talent comes a demand for freedom and subsequently with a lack of control. Talented team players, with careful planning and control are needed to manage the project successfully. –

The Agile manifesto state: “The most efficient and effective method of conveying information to and within a development team is face-to-face conversation”. The studies show that communication in the teams is effective and functions well. Daily standup meetings are common (S2, S5), giving the team a good overview of the tasks the team members are working on and challenges they are facing. The communication with the users on the other hand is not as good and the responsibility to conduct this communication is unclear. The most frequently used methods of conveying information from users to the team are informal (S3, S4, S5). Workshops with team members and users participating are the most frequent method for communication, and informal meetings and evaluations with users are also frequently used (S3, S4). There are examples where the PO takes care of the user communication to protect the team from face-to-face communication with users or customers, for the team the goal is to be able to focus on their development work during the sprint (S5). User forums, blogs, tweets, emails and phone calls with users are easy and simple ways to communicate with users if face-to-face is not possible (S4, S5).

Agile methods have successfully established the tradition of short problem-solving stand-up meetings to save more time for the actual development work (S2, S3, S4, S5). However, often with the effect of lacking or poor documentation. The problem-solving stand-up meetings support the core values of UCSD by building a user-centered attitude and increasing the professional attitudes of the project work.

5.2. Working software

Another Agile principles states: “Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.” According to the studies potentially shippable products are made in two out of three sprints (S2) and the length of the sprints are in general 2–4 weeks (S2, S3). This constrains the UX professional's work and the most frequent reason for not conducting usability evaluation is time constrains (S2, S4). Many IT professionals find it hard to maintain the vision for the entire user experience of the product since the

emphasis is on developing small parts of the final product in each sprint in Scrum (S2, S4). Some of the IT professionals claim that the vision therefore should be stated in the pre-study phase before the actual development starts and then they regularly need to refer back to the project vision to check if it should be changed (S4).

Agile stresses the need for software to be working to manage and facilitate rapid and continuous development. Our research show that the Agile team members interpret that the software is working from a technical and functional perspective, rather than based on the user perspective (S4). This can lead to confusion about when a user story is actually “done” (S4). UCSD on the other hand is much more prototype driven to make sure that the potential software will have the capacity to make sure that it fulfills the needs of the users, since user evaluations rarely fit into the projects (S4).

Making sure that everything is properly functioning is supported by a tight schedule of deliveries of software that needs to run and be testable. It does however require substantial extra work to manage the deployability of the different versions that may mean that quality aspects risks lower priority (S4).

Another principles states: “Working software is the primary measure of progress”. The results show that measures are rarely defined for usability in Scrum projects (S4, S5) and the level of usability of the final product is often informally evaluated with only a few users, to gather qualitative feedback on the current design of the user interface (S4). Few IT professionals conduct quantitative evaluation assessing the level of usability (S4). Formal usability evaluation is highly rated by IT professionals (S3) but even so this way of conducting usability evaluation is not frequently used. The main reason why thorough user evaluation is not conducted is lack of time in the Scrum projects at large (S2, S4). Furthermore, IT professionals that make detailed documentation of usability requirements are not more likely to formally gather feedback from users to check if the documented usability requirements are fulfilled (S4).

5.3. Customer collaboration

The first Agile principle is: “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.” The results from the studies show that developers actually mostly do informal evaluation with users to see if the software is valuable or if the customers or the users are satisfied (S4), even though it could be presumed that developers would evaluate how valuable the software is and how satisfied the customers are, since this is the highest priority. Focusing on the essential functions of the software from the user perspective, prioritizing quality in favor of fully covering the functionality should give sufficient time to meet the expectations.

Many usability professionals acquire usability feedback with user representatives, other project members or user proxies, without going through the rigorous procedures required by many formal usability evaluation methods with actual users (S2, S3, S4). The purpose of the evaluations is most often formative, to receive input on how to improve the system rather than a summative grade assessing the quality of the final product (S4). Some developers commented that they do not know how the system they have delivered is used and do not measure the usage or the satisfaction (S5). Some of the developers presumed that there are some extra features in the system that the users do not need (S5). Sometimes there is confidence that modern technology may solve everything and that it is hard for the users to understand what is best for them (S5).

Another principle related to the qualities of the software is: “The best architectures, requirements, and designs emerge from

self-organizing teams". We have seen that UX requirements are sometimes defined before the actual development starts by business analyst and not the team members (S4). **The requirements relating to usability are rarely described or documented in the projects, as the general idea in Scrum is to focus on communication and not as much on thorough documentation (S4).** This lack of documentation sometimes leads to that the requirements are not evaluated with users, so it is not checked how well the requirements are fulfilled before deploying the developed system to the users (S4).

The Agile manifesto also states that: "Business people and developers must work together daily throughout the project". If the term "business people" is referring to the product owner, we can see in the studies that this principle is often not fulfilled. Developers explain that the product owners are extremely busy and not in contact daily (S4).

Furthermore, we have seen that having daily contact with the customer is hard to accomplish (S5). Mutual learning is an important never ending process and a key factor in UCSD that should be based on mutual understanding between business and development.

Another principle says: "Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely." The studies show that the sprints give the developers a constant pace, because potentially shippable products are in many cases delivered already after two-week sprints (S3). But the pace for involving users is not as obvious and constant. The frequency of involving users in Agile development differs from sprint to sprint (S3), and developers comment that it is hard to know when to gather feedback from the users (S4).

Starting with a higher pace to include users and keeping that pace may allow the Agile professionals to deliver on time keeping the sanity, and even, if the time allows, with improved quality and usability.

5.4. Responding to change

The second Agile principle state: "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage".

Being responsive to change is important both in UCSD and in Agile approaches.

The opportunities to fix defects reported by customers are usually prioritized highly (S5) by Agile professionals, but other changes requested from customers or usability specialists, are not prioritized as highly (S4). One interviewee claimed that usability activities needed to appear as new activities on a post-IT note on the product backlog – otherwise there is a risk that these activities are omitted in the process (S4). The interviewee explained that the team focuses on finishing tasks so the burn down chart will look nice. Working on a story point, that the team though was done is not encouraged by the team, since this does not count as burned story points on the chart (S4).

Changing requirements should be a natural part of iterative development, but it should not mean that Agile professionals should ignore the important preparatory work in understanding the context of use and exploring various design options as emphasized in UCSD. In our experience Agile development processes sometimes neglects the early phases and believes that this can be made up for later on (S4.S5).

The last Agile principle is: "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly". Many of the interviewees in the study take part in retrospective meetings aimed for improving the process for the software development (S2, S5). One informant explained, that

at their retrospective meeting, the team decides on one issue to work on during the next sprint, to extend the likelihood of that issue actually being addressed (S5). Another informant explained that the team started to become tired of hearing the same issues discussed over and over again during the retrospective meetings (S5). Often at the retrospectives, technical things are discussed, like what libraries to use in the development, but rarely how to improve the usability or user experience of the software (S5).

A mechanism needs to be in place to regularly question how the work could be improved to better meet the quality aspects that are so much sought for. This needs to involve the customers and the users who are the ultimate beneficiaries of the developed software.

6. Discussion

One of the bigger challenges in the integration of UCSD and Agile development is the values or attitudes that lie within the different approaches. In Agile development the focus is on speed and communication within the team. In UCSD the emphasis is on communication with users and making systems usable, which may be time consuming. Scientifically the software engineering field does not easily incorporate knowledge from the HCI field without it being expressed using their conceptual framework and following their conditions, often with the consequence that the basic values of UCSD may disappear. For a successful integration of UCSD into Agile development the usefulness of UCSD in Agile processes needs to be made more explicit and the attitude to UCSD needs to develop.

6.1. Implication for the integration of Agile and UCSD

The Agile processes provide a structured approach to group collaboration in software development, focusing on early delivery of functioning code and minimizing "waste" in the process. Using Agile has contributed to the increased likelihood of delivering functioning systems on time and within the budget. According to Mike Cohn's blog the 2012 Standish Group report shows an increased rate of successful projects from 14% in 2002 to 42% in 2010 much attributed to Agile development methods (Cohn, 2012). However, the Agile processes provide little scaffolding when it comes to the actual software development, as it does not describe activities, deliverables, milestones and tools in detail. This is left to the team to take care of based on their experiences and priorities. Nor does it consider the core values of UCSD.

6.2. Does Agile consider the user perspective?

Many Agile methods do not seriously consider the actual end-users; they more often focus on delivering values to the customer than actually delivering quality to the end-users. An example of this is the product owner role. The product owner should express the needs of the user by being responsible for the customer requirements. But, ever so often you see that this role has been shouldered by a member of the software development organization rather than a representative from the user population or the customer, which would have been the case if you would follow a UCSD approach. There is clearly a risk that the product owner could be considered a user substitute. The role of the product owner is a step forward when comparing with traditional approaches where the user perspective was hidden or less explicit. For example, when comparing the Agile approaches with the Rational Unified Process, the concept of the user was substituted by the actor (Balmelli et al., 2006). An actor could actually be a machine actor, a concept that did not even have to mean a human being. But projects that appoint a role as product owner to be

shouldered by someone from the team makes you suspect that the development team is afraid of directly answering to users, or that it would be essential to protect the team from the users, since this may risk the delivery. User stories (Cohn, 2004) is the Agile way of formulating user needs and requirements. User stories date back as early as in eXtreme programming (Jeffries et al., 2001). Initially developers document requirements through user stories, which are basically textual use-case descriptions (Maurer and Martel, 2002). User stories appeared already in the Agile manifesto where the process all began with the creation of “user stories”. The product owner prioritizes what stories to work on in each deliverable increment. The Agile team assesses each story for the particular increment and assigns a cost for each story and then the stories are grouped to a deliverable increment. The team must make a commitment on a delivery date (Beck et al., 2001). According to Sasse and Johnson (1999) user stories are the first artifacts used to describe interactions between the users and the system, but for implementation purposes developers often require something more formal such as Use Cases to explain the user stories better. Use Cases are similar to user stories but different in that they lose most of the context that user stories maintain (Sasse and Johnson, 1999). But user stories are rarely applied as a tool for interacting or collaborating with users. On the contrary it may again be used as a substitute to proper UCSD.

Based on our experiences we recommend that the product owner and at least one member of the team have UCSD qualifications. It is not sufficient that only the product owner or a team member has knowledge about usability activities. The product owner needs to respect the extra time that usability work takes, and the team that works independently needs to prioritize the usability aspects of the product.

6.3. Is UCSD too research oriented and rigorous to become Agile?

The UCSD methodologies are often considered to be too research-oriented, thorough and ambitious to actually fit to the Agile processes. Maybe the reason for this is the usability engineering perspective that early on was prevalent in UCSD. Usability engineering (Mayhew, 1999, Najafi and Toyoshiba, 2008, Royce, 1970) has an overemphasis on measuring usability and using an analysis-design-evaluation cycle to keep on refining and fine-tuning the detailed design to perfection, leaving the actual work tasks and underlying structures fairly untouched. Maybe this has contributed to the overall view that usability professionals rarely contribute to the actual production of program code, but rather conduct research-like studies and spend an enormous amount of time documenting the findings in reports of questionable value to the actual development of the code. Dray and Siegel argues that UX professionals who focus on doing “studies” as opposed to generating designs and products, will always be perceived as peripheral (Siegel and Dray, 2003). The biggest risk of UCSD is that it often falls into the trap of producing analytical reports about usability deficits, rather than contributing to change and innovative design. Moreover, often usability professionals’ aim at working according to rigorously defined usability methods such as usability evaluations, work analysis or user analysis. Our studies show that often this leads to usability work being either done before the actual Scrum project starts in sprint zero, or as a parallel usability project that does not work according to Scrum (Cajander et al., 2013, Larusdottir et al., 2013, Larusdottir et al., 2012).

One interviewee expressed that: “if the Agile people think your methodology has some value to add to their process they will include it in their toolbox, otherwise they just wouldn’t care”. Implicitly the claim means that no UCSD method so far has proven to be sufficiently useful for Agile developers to the extent that they would consider including them in the process. Concurrently, HCI is

suffering from the same attitude problems as UCSD is considered by many software engineers to be too research oriented, lacking the ability to be Agile and to focus on contributing to the actual development of the system or product. He continues: “We need to be pragmatic and realize that our methods are too structured, robust and time consuming to fit into the Agile way of thinking – we either need to declare the battle lost, or adapt our approaches to become more Agile.”

6.4. Integrating Agile and user-centered systems design

The studies indicate that integration of Agile development and UCSD still has quite a few challenges before functioning well together on equal terms in practice. Scrum is the de facto standard, and the organizations market their product or software development saying that they work according to Scrum. Usability professionals work hard to integrate usability activities, and one of the major difficulties is that it is hard to find time to address usability. The circumstances of usability activities are heavily affected by Scrum’ focus on functionality and deliverables instead of quality. Our recommendations for a more successful integration of Scrum and UCSD include a focus on communication both within the team and with users. The successful usability professionals act as boundary spanners in the project, and are carrier of knowledge and competence between users and IT professionals. Successful usability professionals in the empirical studies often focus on adapting their message to the software developers in the teams when they describe the needs of the users. One of the biggest challenges when working with usability activities in Scrum is to find time for user activities in the sprints. Working with Scrum encourages feedback from users in the sprint demos, which is too seldom since weeks can have passed and much functionality might have changed. Moreover, in sprint demos the functioning deliverable is almost done and the possibility to change things is small. In theory it is always possible to throw away malfunctioning deliverables due to user feedback, but in practice this is very seldom done. A recommendation for a more successful integration of UCSD in Scrum is to adapt to the terminology and culture of Scrum. One such example could be to decide a time for certain user communication, as with the sprint demos and sprint retrospectives that are supposed to happen by the end of every sprint or daily Scrum meetings. A possible example could be coining a new activity: “Weekly user meetings”.

A possible way forward when integrating Scrum and UCSD could be to lower the UCSD standard, and make user involvement activities less formal. Metrics and formal usability evaluations through performance measurement or analytical evaluation methods are not suitable in the Agile environment. Instead we would recommend making use of alternative channels for feedback, such as social media, user forums or tweets to include the users. The studies indicate that new communication channels are emerging, and these seem to adapt easily into the spirit of Scrum. Can the approaches work beside one another in an effective way? Most likely the UCSD activities will have little or no effect on the software engineering process unless it clearly contributes to the redesign and to the actual development. Therefore the HCI community needs to further define and discuss the value of usability activities and user involvement in Scrum projects in practice.

7. Final conclusions and contribution

Based on our theoretical analysis on UCSD and Agile, the studies analyzed and the experiences gained we would like to suggest the following general guidance to projects adopting an Agile methodology, such as Scrum, that has the goal to focus on usability and UCSD, sorted under the respective heading:

7.1. Individuals and communication

1. Define the responsibility for Usability and UX for all roles; team members, Scrum master and PO.
2. Team members responsible for Usability and UX should regularly have face-to-face communication with the actual users and at least once during each sprint.
3. Team members should make use of multiple channels for feedback, such as social media, user forums or tweets to include the users in parallel with face-to-face communication.

7.2. Working software

1. State a clear vision for Usability and UX in an early phase and refer back to it regularly to check, if it should be changed.
2. Define measurable goals for Usability and UX and evaluate regularly with users, if the goals are met.

7.3. Customer collaboration

1. In evaluation with users, it should be checked if the system fulfills the user requirements.
2. Evaluations should be conducted regularly to measure how satisfied the users are and how valuable the software is for them - at least every second sprint.
3. Give the person responsible for evaluating Usability and UX a mandate to influence the subsequent project planning - Give them "License to kill"!
4. A communication plan should be established, for the PO, Scrum master and the team to understand the results of the evaluations.
5. The result of the evaluation needs to lead to measures that must be commissioned and followed up.

7.4. Responding to change

1. Define themes for the retrospective meetings and make improving the Usability and UX as one of these themes.
2. Prioritize change requests from users highly, that support a competitive advantage for the users perspective.

The main contribution of this paper is showing that Agile Software development projects do not support UCSD activities simply through the process itself, nor does it prevent it. If you would want to work with UCSD in an Agile fashion the advice presented here under the final conclusions may be a good starting point for the change. In our studies we have seen that Agile practitioners using Scrum, commonly choose one issue to work on for improving their process at the retrospective meetings. Even if these conclusions are based on our studies with Scrum users we clearly see the potential for them to work more generally for all Agile methodologies, thus the generalization. In that spirit we chose one advice that we recommend giving top priority, when working to improve UCSD activities in Agile development: - give the person responsible for evaluating Usability and UX the mandate to influence the subsequent project planning - give them "A license to kill!" When they have got the authority to "stop the production line", similarly to what is recommended in lean development, the software developers may feel the need for the other improvements suggested above.

Acknowledgments

We wish to thank all participants in the interview studies for their time and open responses to our questions. We wish to thank all previous reviewers of earlier versions of this paper for their constructive comments. Finally we wish to thank our universities

for their support in providing us with the time to conduct this research.

References

- Balmelli, L., Brown, D., Cantor, M., Mott, M., 2006. Model-driven systems development. *IBM Syst. J.* 45 (3), 569–585.
- Bannon, L.J., 1995. From human factors to human actors: the role of psychology and human-computer interaction studies in system design. *Human-Computer Interaction*. Morgan Kaufmann Publishers Inc, pp. 205–214.
- Baxter, G., Sommerville, I., 2011. Soci-technical systems: from design methods to systems engineering. *Interact. Comput.* 23 (1), 4–17.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., 2001. *The Agile manifesto*. The Agile Alliance.
- Blomkvist, S., 2006. *User-Centred Design and Agile Development of IT Systems* Doctoral degree. Uppsala University.
- Boehm, B., 2003. Using risk to balanced agility and discipline: a quantitative analysis. In: *EDSER-5 Conference*.
- Boivie, I., 2005. *A Fine Balance: Addressing Usability and Users Needs in the Development of IT Systems for the Workplace* Doctoral degree. Uppsala University.
- 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.
- 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.
- Cajander, A., Boivie, I., Gulliksen, J., 2008. Usability and users' health issues in systems development - attitudes and perspectives. In: Law, E., Hvannberg, E., Cockton, G. (Eds.), *Maturing Usability*. Springer, pp. 243–266.
- Cajander, A., Larusdottir, M.K., Gulliksen, J., 2013. Existing but not explicit - the user perspective in Scrum projects in practice. *INTERACT 2013*. Springer.
- Chamberlain, S., Sharp, H., Maiden, N., 2006. Towards a framework for integrating Agile development and user-centred design. *Extreme Programm. Agile Process. Softw. Eng.* 143–153.
- Cockburn, A., Highsmith, J., 2001. Agile software development: the people factor. *Computer* 11, 131–133.
- Cohn, M., 2004. *User Stories Applied: for Agile Software Development*. Addison-Wesley professional.
- Cohn, M., 2012. Agile succeeds three times more often than waterfall. <http://www.mountaingoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall>.
- Constantine, L.L., 2002. Process agility and software usability: toward lightweight usage-centered design. *Inf. age* 8 (8), 1–10.
- Constantine, L.L., Lockwood, L.A., 2003. Usage-centered software engineering: an Agile approach to integrating users, user interface and usability into software engineering practice. In: *Proceedings of International Software Engineering Conference*. IEEE Computer Society, pp. 746–747.
- Da Silva, T.S., Martin, A., Maurer, F., Silveira, M.S., 2011. User-centered design and Agile methods: a systematic review. In: *AGILE (2011)*, pp. 77–86.
- Ferreira, J., Sharp, H., Robinson, H., 2011. User experience design and Agile development: managing cooperation through articulation work. *Softw.: Pract. Exp.* 41 (9), 963–974.
- Greenbaum, J., Kyng, M., 1991. *Design at Work: Cooperative Design of Computer Systems*. Laurence Erlbaum Associates, Hillsdale.
- Gulliksen, J., Boivie, I., Göransson, B., 2006. Usability professionals-current practices and future development. *Interact. Comput.* 18 (4), 568–600.
- Gulliksen, J., Boivie, I., Persson, J., Hektor, A., Herulf, L., 2004. Making a difference: a survey of the usability profession in Sweden. In: *Proceedings of NordiCHI 2004*. ACM Press, pp. 207–215.
- Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., Cajander, A., 2003. Key principles for user-centred systems design. *Behav. Inf. Technol.* 22 (6), 397–409.
- Hardenborg, N., 2007. *Designing Work and IT Systems: A Participatory Process that Supports Usability and Sustainability* Doctoral degree. Uppsala University, Uppsala.
- Iivari, J., Iivari, N., 2006. Varieties of user-centeredness. In: *Proceedings of the HICSS'06 Conference*, 8. IEEE.
- ISO, 2010. ISO 9241-210: Ergonomics of Human-System Interaction - Part 210: Human-Centred Design Process for Interactive Systems. International Organisation for Standardization, Geneva, Switzerland.
- Jakobsen, C.R., Johnson, K.A., 2008. Mature Agile with a twist of CMMI. In: *Proceedings of the Agile '08 Conference*. IEEE, pp. 212–217.
- Jeffries, R., Anderson, A., Hendrickson, C., 2001. *Extreme programming installed*. Addison-Wesley Professional.
- Jia, Y., Larusdottir, M.K., Cajander, Å., 2012. The usage of usability techniques in Scrum projects. In: *Proceedings of the HCSE 2012 conference in Toulouse, October (2012)*.
- Kammersgaard, J., 1990. Four different perspectives on human computer interaction. In: Preece, J., Keller, L. (Eds.), *Human-Computer Interaction*. Prentice-Hall university press, Cambridge.
- Kollmann, J., Sharp, H.L., 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.

- Kuusinen, K., Väänänen-Vainio-Mattila, K., 2012. How to make Agile UX work more efficient: management and sales perspectives. In: Proceedings of NordiCHI 2012 Conference. ACM Press, Copenhagen, Denmark, pp. 139–148.
- Larusdóttir, M.K., Bjarnadóttir, E., Gulliksen, J., 2010. The focus on usability in testing practices in industry. In: Proceedings of the Human Computer Interaction Symposium at the World Computer Congress 2010. Brisbane, Australia.
- Larusdóttir, M.K., Cajander, Å., Gulliksen, J., 2013. Informal feedback rather than performance measurements – user centred evaluation in Scrum projects. *Behav. Inf. Technol.*
- Larusdóttir, M.K., Cajander, Å., Gulliksen, J., 2012. The big picture of UX is missing in Scrum projects. In: Proceedings of the Ix-Used Workshop at NordiCHI, October.
- Larusdóttir, M.K., Cajander, Å., Simader, M., 2014. Continuous improvement in Agile development practice: the case of value and non-value adding activities. In: Proceedings of the HCSE Conference. Paderborn.
- Larusdóttir, M.K., Haraldsdóttir, O., Mikkelsen, B., 2009. User involvement in Icelandic software industry. In: Proceedings of the 2nd international workshop I-Used 2009, at the INTERACT conference. Sweden.
- Larusdóttir, M., Law, E., Cajander, Å., 2015. User and client satisfaction in Agile development. In: INTERACT 2015 Adjunct Proceedings: 15th IFIP TC. 13 International Conference on Human-Computer Interaction 14–18 September 2015, 22. Bamberg, Germany. University of Bamberg Press, p. 249.
- Law, E.L.C., Lárusdóttir, M.K., 2015. Whose experience do we care about? Analysis of the fitness of Scrum and Kanban to user experience. *Int. J. Human-Comput. Interact.* 31 (9), 584–602.
- Maurer, F., Martel, S., 2002. Extreme programming. Rapid development for web-based applications. *Internet Comput.* 6 (1), 86–90.
- Mayhew, D.J., 1999. *The Usability Engineering Lifecycle*. Morgan Kaufmann Publishers, San Francisco, USA.
- McInerney, P., Maurer, F., 2005. UCD in Agile projects: dream team or odd couple? *Interactions* 12 (6), 19–23.
- Najafi, M., Toyoshiba, L., 2008. Two case studies of user experience design and Agile. In: Proceedings of the Agile '08 Conference. IEEE Computer Society.
- Nurminen, M.I., 1987. Different perspectives: what are they and how can they be used?. In: Docherty, e.a. (Ed.), *System design for Human Development and Productivity: Participation and Beyond*. Elsevier Science Publisher B. V.
- Nygaard, K., 1990. Program development as a social activity. In: Proceedings of the Participatory Design Conference.
- Olsson, H.H., Alahyari, H., Bosch, J., 2012. Climbing the “Stairway to Heaven” – a multiple-case study exploring barriers in the transition from Agile development towards continuous deployment of software. In: Proceedings of the 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), (2012). IEEE, pp. 392–399.
- Orlikowski, W.J., Gash, D.C., 1994. Technological frames: making sense of information technology in organizations. *ACM Trans. Inf. Syst. (TOIS)* 12 (2), 174–207.
- Persson, J., 2004. *The Obvious and the Essential. Interpreting Software Development and Organizational Change* Doctoral degree. Uppsala University, Uppsala.
- Rosson, M.B., Carroll, J.M., 2001. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufmann.
- Royce, W.W., 1970. Managing the development of large software systems: concepts and techniques. In: Proceedings of IEEE WESCON Conference.
- Salah, D., Paige, R.F., Cairns, P., 2014. A systematic literature review for Agile development processes and user centred design integration. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. ACM.
- Sasse, M.A., Johnson, C., 1999. How stories capture interactions. In: Proceedings of INTERACT'99. Ios Press Inc.
- Schwaber, K., 2009. *Agile Project Management with Scrum*. Microsoft Press.
- Schwaber, K., Beedle, M., 2002. *Agile Software Development with Scrum*. Prentice Hall, Upper Saddle River, N. J.
- Sefelin, R., Tscheligi, M., Giller, V., 2003. Paper prototyping – what is it good for? A comparison of paper- and computer-based low-fidelity prototyping. In: Proceedings of CHI'03 Conference on Extended Abstracts. ACM.
- Siegel, D.A., Dray, S., 2003. Living on the edges: user-centered design and the dynamics of specialization in organizations. *Interactions* 10 (5), 18–27.
- Sommerville, I., 1996. Software process models. *ACM Comput. Surv. (CSUR)* 28 (1), 269–271.
- Svanæs, D., Gulliksen, J., 2008. Understanding the context of design: towards tactical user centered design. In: Proceedings of NordiCHI 2008. ACM Press, pp. 353–362.
- Sy, D., 2007. Adapting usability investigations for Agile user-centered design. *J. Usability Stud.* 2 (3), 112–132.
- Sy, D., Miller, L., 2008. Optimizing Agile user-centred design. In: Proceedings of CHI'08 Conference on Extended Abstracts. ACM, pp. 3897–3900.
- Takeuchi, H., Nonaka, I., 2016. The new new product development game. *Harv. Bus. Rev.* 64 (1), 137–146.

Marta Lárusdóttir is an assistant professor in the School of Computer Science at Reykjavik University, Iceland. Her main research area is the integration of the user perspective in software development, particularly focusing on researching user centered evaluation both in practical and experimental settings. Furthermore, Marta has focused researching the integration of the user perspective in Agile, and studied Scrum particularly. Marta is the chair for the CRESS research center at Reykjavik University.

Jan Gulliksen is a professor in Human Computer Interaction at KTH Royal Institute of Technology, Sweden. He also holds the position as Dean of the School of Computer Science and Communication at KTH. In 2012, Jan was selected to be “Sweden’s Digital Champion” by the Swedish Government, and he is also Chairman of the Government’s Digital Commission. His recent research interests are related to UCD and especially Agile development processes.

Åsa Cajander is associate professor at the Department of Information Technology, Uppsala University, Sweden. Åsa has focused on usability of IT in government organizations and health care in her research, and on teaching and learning of HCI. In the former area she have been an action researcher in a large number of projects focusing on software development processes, and especially on user centered systems design and Agile development processes. In the latter area, Åsa’s research has focused on improving education for students in the field of HCI.