



Integrating UX work with agile development through user stories: An action research study in a small software company

Alisa Ananjeva^{*}, John Stouby Persson, Anders Bruun

Aalborg University, Computer Science, Selma Lagerlöfs Vej 300, DK-9220 Aalborg Øst, Denmark

ARTICLE INFO

Article history:

Received 19 December 2019
Received in revised form 1 July 2020
Accepted 12 August 2020
Available online 21 August 2020

Keywords:

User experience
Agile software development
Action research
User story
Artifact

ABSTRACT

The integration of user experience (UX) work with agile software development has been addressed in extensive research of challenges and process models. However, in-depth research of context-specific improvements of this integration with actual UX professionals and agile developers in their actual practice is limited. This study examines how the integration of UX work with agile development can be improved in the context of a small Danish Software-as-a-Service (SaaS) company. We used the problem- and solution-oriented action research method over 12 months in the company. During this period, we initially carried out extensive participant observations, recorded 32 semi-structured interviews, and finally conducted two improvement iterations with evaluations of their effect on agility. We identified user stories as an essential indicator of UX integration. Verbose user stories imply problems in collaboration and trust, while concise user stories and deliberation improve integration of UX work with agile development. The conclusion is that integrating UX work with agile development in practice is complex, contextualized, and difficult even for only a small part of it. We propose that concise user stories and deliberation can be useful and well-defined focuses for integrating UX work with agile software development without sacrificing their agility.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

During the last two decades, both agile development and user experience (UX) work have become standard practices in the software industry (Brhel et al., 2015; Bruun et al., 2018; Chilana et al., 2010; Larusdottir et al., 2017). Both domains strive to build quality software (Ferreira et al., 2011) and contribute to economic success in highly competitive markets (Brhel et al., 2015). However, agile development and UX work utilize different approaches, values, and views on what quality software is (Ferreira et al., 2011; Larusdottir et al., 2017). Even though both domains are iterative, the rigor of UX up-front activities clashes with the rapidness of agile development (Larusdottir et al., 2017), which imposes a challenge for integrating UX with agile development (Chilana et al., 2010; Ferreira et al., 2011).

UX integration has been an area of interest for the academic community for over a decade (Brhel et al., 2015) and has generated numerous studies of processes, challenges, and success factors (Brhel et al., 2015; da Silva et al., 2013; Kuusinen et al., 2012). These studies are valuable in gaining an understanding of the current state of UX integration; however, they have captured only a glimpse of a practice, lacking an understanding of events

that have an effect on the transformation of a practice (Kashfi et al., 2019).

One of the main challenges within UX integration is to build a common ground between UX professionals and agile developers (Garcia et al., 2019). Artifacts, such as prototypes and user stories, aid the establishment of a common ground and a common understanding (Brhel et al., 2015; Garcia et al., 2019; Jones and Thoma, 2019). Artifacts are boundary objects between UX professionals and agile developers, and are used during collaborative activities (da Silva et al., 2018; Garcia et al., 2019). Artifacts are therefore seen as a fundamental part of the software development process (Zaitsev et al., 2020). A user story is a popular artifact within software development (Lucassen et al., 2016). The user story method stems from agile discipline and is used to describe user values and needs (Cohn, 2004). A user story is short, comprehensible, and negotiable in order to mediate information to any kind of stakeholder regardless of their department, educational background, or technical insight (Cohn, 2004). Thus, user stories are beneficial for the interdisciplinary work.

Generalizable and rational solutions are not effective because these are rarely followed in practice due to local circumstances within each organization (Ferreira et al., 2011). Thus, UX integration research needs a more nuanced view of how user stories can be useful in practice, which requires long-term studies (Kashfi et al., 2019) that emphasize the situated practice in which the issue of UX integration is embedded (Ferreira et al., 2011). A

^{*} Corresponding author.

E-mail addresses: alisaa@cs.aau.dk (A. Ananjeva), john@cs.aau.dk (J.S. Persson), bruun@cs.aau.dk (A. Bruun).

case study, a prevalent research method used when studying UX integration with agile development, makes it difficult to uncover what aspect of a practice improves UX integration (da Silva et al., 2015). Thus, researchers ought to make interventions that are tailored for the local practice (Mckay and Marshall, 2001).

We contribute to the calls for research (Kashfi et al., 2019) with an action research (AR) study (Hayes, 2011; McKay and Marshall, 2001) conducted in a small Danish Software-as-a-Service (SaaS) company – ServiceSoft. Our aim was to determine how we can improve the integration of UX work with agile software development. The use of *we* implies not only we as researchers but also the practitioners, thus emphasizing the collaborative nature of AR. The verb *improve* refers to change, which is a core principle of AR, to benefit practice and contribute to research (McKay and Marshall, 2001). Integration implies that we recognize UX and agile software development as distinct but mutually dependent practices and areas of research. Segregating them into two independent areas would undermine their interdependent relationship in making useful software (Detweiler, 2007; Ferreira et al., 2011). Thus, in this study, we as researchers collaborated with practitioners on making changes that improve the integration of UX work with agile software development while maintaining the distinctive strengths of the two practices.

In the following, we first contextualize our study in related research. Afterward, we describe our AR approach to address our research question and the problematic situation in a specific company, ServiceSoft. ServiceSoft is a pseudonym used throughout the article to ensure the anonymity of the company. In the findings section, we present our two interventions at ServiceSoft and note what consequences they had on the agility of its practice. Finally, we discuss the contribution of our findings, limitations, followed by a conclusion.

2. Related research

In the following section, we appreciate the problem of UX integration in agile software development through related research. In Section 2.2, we present user stories, and discuss artifact mediated communication as a possible solution to the problem of UX integration.

2.1. Integration of UX in agile software development

Agile developers value rapidity, flexibility, adaptability, lean-ness, and customer-centricity (Conboy, 2009; Persson et al., 2018). UX designers value in-depth up-front activities with a purpose of understanding the end-user and the context of use (Bruun et al., 2018; Larusdottir et al., 2017) and rigorous user evaluations (Brhel et al., 2015). Despite these differences, integration of UX with agile development is essential for introducing the users' point-of-view in software development without undermining their agility. We do not consider agility to be binary or something a company or a project group either is or is not. Agility is a dynamic continuum – a combination of independent and interdependent conditions – and is broadly understood as the “continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change and learn from change while contributing to perceived customer value (economy, quality and simplicity) through its collective components and relationships with its environment” (Conboy, 2009, p. 340).

The literature regarding UX integration is immense, and contributes to an understanding of the challenges, success factors, and best practices for merging the two domains. A majority of the research regards processes and techniques, which are more mature aspects of UX integration (Brhel et al., 2015). A Cycle

Zero process (Sy, 2007), also referred to as Sprint Zero (Wale-Kolade et al., 2014), is often purposed as a possible solution to UX integration (Brhel et al., 2015). Cycle Zero is a parallel and interwoven development process during which UX professionals work a sprint ahead of agile developers (Sy, 2007). Little Design Up-Front (LDUF) is an agile adaptation of in-depth up-front activities (Kuusinen et al., 2012; da Silva et al., 2013). However, processes and techniques are rarely methodically followed in practice due to the highly contextualized nature of software development, which can make them ineffective (Fitzgerald, 1997; da Silva et al., 2018). Rational plans and solutions (Ferreira et al., 2011) often do not consider organizational settings (Kashfi et al., 2019), nor communicative aspects (Brhel et al., 2015), of UX integration. Barksdale and McCrickard (2012) have argued that UX integration is a question of establishing a shared practice. However, there are multiple views on how to achieve such shared practice. Wale-Kolade et al. (2014) suggested that UX professionals must have skills in front-end development, while other research (Bruun and Stage, 2015) has suggested training agile developers in basic UX methods. Both approaches argued that obtaining skills of the other domain helps bridge the gap between the UX professionals and agile developers.

In a comprehensive literature review of UX integration with agile development, Brhel et al. (2015) found a lack of research regarding collaborative aspects of UX integration which are important for establishing a shared practice (Barksdale and McCrickard, 2012). Collaboration is important in the early stages of software development for creating a common understanding of the problem and detailing the design and implementation (Brown et al., 2011) to achieve a common goal (Barksdale and McCrickard, 2012). However, the complex and interdependent aspects of social interactions require cognitive attributes regarding how a team learns and shares knowledge and best practices, as well as relational attributes concerning trust (Barksdale and McCrickard, 2012). Lack of trust can hinder the emergence of knowledge; if UX professionals “believe that sharing practices with agile software developers will be used against them in some way, they may be less inclined to share that knowledge” (Barksdale and McCrickard, 2012, p. 68). The issue of trust goes both ways; the agile developers also tend to distrust UX practices (Barksdale and McCrickard, 2012; Bruun et al., 2018; Chilana et al., 2010). To obtain mutual trust both domains must be willing to compromise, learn, and share knowledge (Barksdale and McCrickard, 2012; Wale-Kolade et al., 2014). When sharing information, written specifications and requirements cannot stand alone (Cohn, 2004). Verbal communication is required in order to efficiently share new knowledge (Cohn, 2004; Melnik and Maurer, 2004). Thus, in order to obtain mutual trust, verbal communication is essential.

Sketches, mock-ups, prototypes, and user stories are central means of communication for UX professionals and agile developers (Brhel et al., 2015; Garcia et al., 2019). These artifacts facilitate communication and support collaboration between the two domains (Brown et al., 2011; Garcia et al., 2019; Paay et al., 2009) and help teams to reach an agreement regarding a product (da Silva et al., 2018). In the early stages of software development, artifacts are fundamental to support information sharing, focus the ideation process (Brhel et al., 2015; Brown et al., 2011; Garcia et al., 2019), and achieve a common understanding of a problem (Brown et al., 2011). However, there is a scarcity of studies that examine whether artifacts facilitate emergence of mutual trust between the two domains in practice (Garcia et al., 2019), which has been established as one of the key factors that positively affects integration of UX with agile development (Barksdale and McCrickard, 2012).

2.2. User stories

A user story is a method to describe the functionality of a software, with an emphasis on user values and needs (Cohn, 2004) and stems from the domain of agile software development. A user story is often written in the format “As a [type of user], I want [some goal] so that [some reason]”. The aim is to keep the story short and comprehensible to any stakeholder involved, which makes a user story a link among different types of stakeholders, regardless of department, educational background, or technical insight. Thus, this format enables dissemination of knowledge between UX professionals and agile developers.

User stories consist of three dimensions: card, conversation, and confirmation (Cohn, 2004). The research on user stories most often involves the artifact – the *card* – focusing on syntax, semantics, and the pragmatic properties of the user stories (Brhel et al., 2015; Lucassen et al., 2015), that is, the user story evaluation tool INVEST, which supposes that a good user story is Independent, Negotiable, Valuable, Estimatable, Small and Testable (Lucassen et al., 2015, 2016), and user story templates

(Lucassen et al., 2015). The *confirmation* is also a frequently mentioned dimension; it is deemed to further the integration of UX with agile development when used as an evaluative tool, for instance, during usability testing (da Silva et al., 2013; Peres et al., 2014) or acceptance tests (Cohn, 2004; Lucassen et al., 2015). The *conversation*, however, is mostly overlooked by the academic community. Even though it has been established that artifacts support collaboration (Garcia et al., 2019) and coordination (Zaitsev et al., 2020), few studies have been conducted regarding how a card or an artifact should further a collaborative practice between the two domains and how it contributes to emergence of mutual trust. Lucassen et al. (2016) studied the effectiveness of user stories in practice, and found that practitioners experience that user stories further a common understanding of the requirement, which consequently leads to the development of the *right* software. However, they do not describe how the process of the conversation furthers the collaboration and how the knowledge is being spread. We address the identified research shortcomings through AR in a small software company, ServiceSoft, as detailed in the following section.

3. Methodology

To further our understanding of how we can improve the integration of UX work with agile software development, we used an AR method in “the seeking of solutions or improvements to ‘real-life’ practical problem situations” (Mckay and Marshall, 2001, p. 47). AR is a problem- and solution-oriented method of initiating a sustainable change that emphasizes both theory and practice with the objective to apply, as well as add to, the existing body of knowledge (Mckay and Marshall, 2001), in this case on the integration of UX with agile software development. Following an iterative process encouraging relevance in a situated practice (Avison et al., 1999; McKay and Marshall, 2001), we collaborated with ServiceSoft, a Danish SaaS company, over 12 months. As stated by Ferreira et al. (2011, p. 972) “[o]nly an examination of the values and assumptions within their organizational setting brings an understanding of how [practitioners] work and why”. AR focuses on understanding “what the practitioners do, not what they say they do” (Avison et al., 1999, p. 96) and on real-life problems to avoid “the potential separation of research and practice” (Mckay and Marshall, 2001, p. 48). The first author carried out participant observations and *ad hoc* conversations, and recorded 32 semi-structured interviews (Myers and Newman, 2007) to collaboratively formulate the problem (Nielsen

and Persson, 2016) of UX integration with ServiceSoft. Research activities are elaborated in Section 3.2. Moreover, we conducted two problem-solving cycles involving ServiceSoft employees and managers to ensure relevance and collaborative learning of the situation.

3.1. The company – ServiceSoft

ServiceSoft is a small Danish SaaS company which creates cloud-based software for meeting and workplace management. The most popular product is a Microsoft Outlook (Microsoft, Redmond, WA, USA) add-in. This Outlook add-in enables users to find all relevant resources, such as meeting rooms and hot desks, across locations and time zones. It also indicates their status at the time of the meeting and displays available facilities at the site. At the time of the study, the company had six meeting and workplace management software products and an app under development.

The company has two locations with their main office in Denmark, and a smaller office in Boston, Massachusetts in the USA with a total of 28 employees in the two locations. The mission, as well as the vision, of the company is to revolutionize workspace and enable a modern workplace environment. At the time of the study, the company had over 150 enterprise customers and more than 200,000 active users worldwide. ServiceSoft has various user groups such as secretaries, canteen employees, and administrative staff. In order to accommodate the broad needs of enterprise customers, each product is customized for each group. This requires an in-depth understanding of each group and its activities. The responsibility for product development is distributed between two teams: the product team (PT), which is responsible for the UX work, and the development team (DT), which represents agile software development. The study was conducted at the main office, where the PT and the DT are located. At the time of the study, the DT had four developers, who describe themselves as generalists, even though they all have areas of expertise and specific responsibilities (e.g. front-end or back-end development). The head of development is a former senior developer at ServiceSoft. He became head of development due to his fifteen years' experience with the integration of agile development processes, and he is in charge of agile transformation. His responsibilities involve dissemination of agile values throughout the DT and adaptation of Scrum (Schwaber and Beedle, 2002), which is a popular agile project framework (Larusdottir et al., 2017).

At the beginning of the study, the product manager was responsible for most of the PT's activities. However, to make the product more user-centered, the PT was expanded to four members: product owner, product manager, a user on-boarding specialist, and product expert. All members of the PT have an extensive knowledge regarding the domain, customers, end users, and have an insight into business development. The PT follows Scrum principles initiated by the DT, and use an adjusted version of Cycle Zero (Sy, 2007). They are responsible for the up-front activities, such as understanding the user, obtaining domain knowledge, information hierarchy, some visual design, development of user stories, and testing software. The PT aspires to carry out field work in order to meet the users in their working environment, and get insights directly from the users. However, due to a lack of resources, field work is not always possible in the context of ServiceSoft. Therefore, in order to carry out the UX work, the PT communicates with other departments, such as Marketing and Customer Success Management (CSM). The PT mostly collaborates with the DT, regarding technical limitations and possibilities for what is possible to implement. The DT and the PT share information through Azure DevOps (Microsoft, Redmond, WA,

USA) which is an agile development management tool. Furthermore, the PT has an intermediary role in making sure that all departments know of new features, new products, their business value, and what customer problems are being solved and how. Communicating with the different departments, gathering and disseminating the same information many times without compromising it is a challenging for the PT. Thus, the PT is not only required to collaborate and communicate with the DT while sustaining its agility, but also to enable the other departments to do their job.

ServiceSoft's transformation towards becoming agile and user-centered, considering that both domains have been recently established, makes it an interesting setting for studying how to improve the integration of UX work with agile software development. AR theory (Mckay and Marshall, 2001) states that the improvement of a problem situation requires a targeted intervention based on an in-depth understanding of the theoretical and real-world context in which the problem occurs. In the following section, we describe our research activities and how these contributed to our understanding of the problem situation.

3.2. Research activities

To gain an in-depth understanding of the problem situation at ServiceSoft, the first author conducted participant observations (Lazar et al., 2017). The first author participated in the day-to-day activities of the PT from August to December, 2018. During that time, she was able to gain firsthand experience working with both the PT and the DT (see Fig. 1). The first author had multiple ad hoc conversations with both the PT and the DT and also recorded 32 semi-structured interviews in accordance with the guidelines described in Myers and Newman (2007).

The 32 recorded interviews were distributed between all four developers, head of development, and three PT members (see Fig. 1). During the study, sections of the interviews that are relevant to the research question were selectively transcribed (McLellan et al., 2003). We used the selectively transcribed interviews to capture, and substantiate, breakdowns (astonishment or mysteries) in our understanding of the problem situation (Brinkmann, 2014).

An example of this approach occurred during the first intervention; the product manager shared how she felt when the DT responded to a user story (see Section 4.2). We captured this breakdown with the quote: "you tell me this, but yet I don't believe you and I will challenge you no matter what you say". (Product Manager). Through this statement, we found that a user story is not just a representation of user needs, it is also a UX professional's work and pride. Similar to this example, every breakdown nuanced our understanding of the problem and the solution, which guided our inquiry.

The aim of the inquiry with the semi-structured interviews was decided in biweekly debriefings (Spall, 1998) with the second and third authors. The interview data were used as a shared point of reference between the authors in substantiating the first author's observations and experiences at ServiceSoft. The first author is a Master of Science in Informatics focusing on Computer Science, Software Engineering and Communication studies. The second author is an associate professor in Computer Science with experience in conducting qualitative studies and action research of agile software development over the past 10 years. The third author has a prior experience working as a UX professional, and his current research concerns theoretical, methodological, and practical aspects of UX. Both the second and the third authors contributed with an in-depth knowledge, and experience, in regards to agile software development, User Experience practice, and integration of both domains. Together we identified collaborative issues between the UX professionals and agile developers

at ServiceSoft. By studying the two domains as two separate but interconnected entities we inquired as to how the two domains perceive themselves and each other, and how they collaborate in order to understand the problem situation. Furthermore, to determine a potential contribution to the academic community and to identify a problem-solving approach, we related the problem situation to the existing body of knowledge.

Our initial understanding of the problem situation was that the DT lacked user understanding, and the PT was overworked by time-consuming up-front activities (see Fig. 1). Initially, ServiceSoft was interested in this intervention, however, due to a lack of resources and a direct access to the end user this intervention was found to be unattainable. We pivoted our study focusing on what is actionable in the context of the company. Eventually we determined that the collaborative issues were particularly manifest in their use of the user story method (see Fig. 1) as elaborated in the section on our findings. The PT developed verbose user stories, which was in opposition to the agile value of less documentation and more frequent communication (Cohn, 2004; Conboy, 2009) between the two domains. In the context of ServiceSoft, verbose user stories, albeit an artifact, limited the conversation dimension of a user story, thus reducing collaboration and knowledge dissemination between the two domains.

To initiate change and solve the problem situation of UX integration with agile development, we initiated a targeted intervention which was based on the established knowledge that a user story must be concise (Cohn, 2004). Through two iterations (see Fig. 1), we were able to initiate change at ServiceSoft and acquire knowledge on how an improvement can be introduced for UX integration without undermining the agility of the software development practice. Agility is undermined when time-consuming UX work, for instance, rigorous up-front activities and usability evaluations, hinders the rapidness of agile software development practice (Larusdottir et al., 2017). However, UX activities are important in order to ensure that actual user experiences become an integrated part of the development process. It is essential to integrate UX work with agile software development, without undermining the agility of both the UX and the software development practices (Persson et al., 2018). We evaluated the impact of our intervention on the agility of the development practice according to Conboy's (2009, p. 341) three-point taxonomy of agility. First, to be agile, a method component *must* contribute to one or more of the following: (i) creation of change, (ii) proaction in advance of change, (iii) reaction to change, and (iv) learning from change. Second, to be agile, a method component *must* contribute to one or more of the following and must not detract from any: (i) perceived economy, (ii) perceived quality, and (iii) perceived simplicity. Third, to be agile, a method component *must* be continually ready, that is, requiring minimal time and cost to prepare the component for use. Conboy's taxonomy of agility is used to analyze the agility of the software development practice and the UX work (Persson et al., 2018).

4. Findings

In this section, we present the problem situation at ServiceSoft and our two interventions to improve the integration of UX work with their agile software development.

4.1. The problem situation at ServiceSoft – Verbose user stories

In January 2019, ServiceSoft set the goal of making their product more user-centered, while the DT initiated an agile transformation towards less documentation and more incremental software development. These changes limited the up-front activities of the PT. However, the DT started to experience the

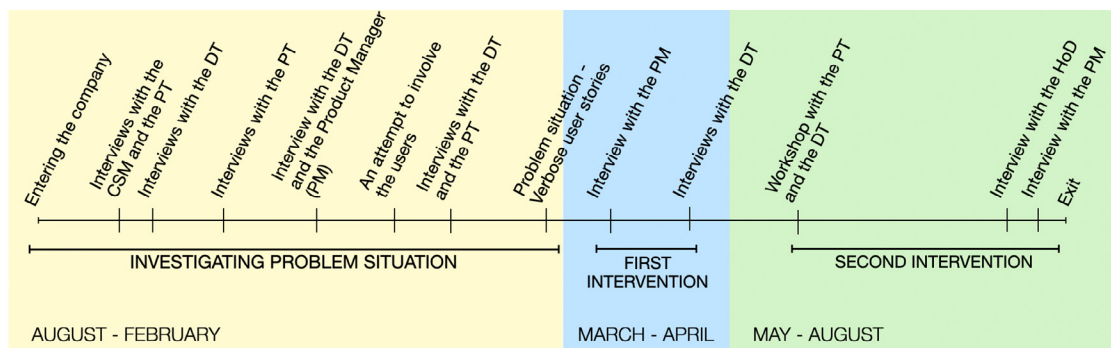


Fig. 1. Research activities throughout the AR study.

situation as they were “doing the product team’s job”, while the PT “could not understand that things were not properly delivered”. The DT became resistant to making changes in the product to avoid *wasteful work*, which manifested a distrust towards the PT and their activities. The DT began to question the validity of the presented user stories, and multiple user stories were being dismissed. When the PT experienced this distrust towards their professional competences, they started to use a lot of time on writing long and elaborate user stories to increase their credibility. Verbose user stories became the PT’s shield to protect them from the DT.

I really feel sometimes that I have to go in front of the lions. [You] would be eaten alive if you are not saying the right thing, if you are not communicating in a right way, if you do not have exactly your value proposition on point, and if you cannot argue precisely and very short then it can be very, very scary to actually come and try to discuss these user stories. That might also be a reason for why I need to prepare myself so much. It might be overdone sometimes.

[User On-boarding Specialist - PT]

The user on-boarding specialist describes a stressful environment where there is no room for mistakes or ambiguity. She describes the DT as “lions” ready to tear a user story apart. The user on-boarding specialist writes verbose user stories to protect herself and a user story — even “overdone sometimes” in reference to its length. A PT member writes a verbose user story in a close collaboration with the CSM team. The CSM team has a direct access to the end users and their feedback, concerns, wishes and experiences with the product. A PT member applies these insights when creating a verbose user story. The process of writing a verbose user story takes place weeks prior to its delivery and usually contains context descriptions, possible solutions, ideas, and known limitations. However, the DT regarded such *possible* solutions, ideas, and limitations as *final*. Both teams were unaware of each other’s challenges, which undermined the teams’ ability to empathize with each other. The DT felt that the PT took ownership over *how* a product must be developed and not only *what* must be done and *why*.

It is a quite important distinction. Because, if you own “what”, then you decide what the product should do, but how much of the “how” should you own? Because, we believe that we should own “how”, but “how” can have some UX implications. Because, we can say that this is the smartest way of solving a task, however, if it does not fit the flow of the product then we have a problem. At the same time, it is an issue if the Product Team owns “how”, because they can think of a solution that simply cannot be done. We are bound by technological limitations.

[Front-end Developer - DT]

The developer acknowledges that the ownership of “how” is important for both teams. The DT and the PT members preferred clear responsibilities. They did not consider joint ownership of how a product must be developed as a possibility. Both teams defended their ownership of “how” the product must be developed. Against this backdrop, the head of development and the product manager introduced *grooming sessions* meant to be a creative outlet and a collaborative tool to involve both the DT and the PT in the exploratory activities. However, power struggles over *how*, invoked by verbose user stories, became more evident during the grooming sessions due to, at times, heated discussions regarding solutions or semantics:

Once [during a grooming session] I had a question regarding something I was confused about, something about a naming. To my question I got a response “What do you think?” My pulse was 120, because I thought that it was disrespectful, and I do not have time for this kind of nonsense. I asked the question for a reason. It was not a rhetorical question. And I wanted to say “What do YOU think?” If I ask a question, it is because I do not know the answer, and I do not expect to get that kind of response back. This is where it has been a challenge for me, that there has not been room for exploration. There has not been room for those who do not understand a problem 100%. They are not invited. Because, when you ask a question, people start to roll their eyes. You attend these grooming sessions with the feeling that one or two people are more or less allowed to decide how to do it. And I think this is a huge problem.

[Front-end Developer - DT]

The developer experienced being excluded due to lack of knowledge, which reinforced defensive behavior. Defensive behavior often occurs during collaboration between the two teams. Both the PT and the DT were aware of the need for collaboration. However, defensive behavior obstructs the dissemination of knowledge and the advancement of a collaborative practice. Both teams “have been talking about collaboration” and “there being a collaboration”, nevertheless “there really has (not) been much collaboration” (Back-end Developer - DT). The back-end developer presented the main challenge that ServiceSoft experienced while integrating UX work with agile software development. Despite targeted actions, ServiceSoft was unable to improve the collaboration between the two teams. The defensive attitudes of the PT and the DT, albeit due to different reasons, resulted in unsuccessful discussions concerning the DT and the PT and how they ought to collaborate. Both teams promoted their own agenda, and disregarded the values that the other team represented. The back-end developer stated that “We do not involve them, and they do not involve us”, suggesting an “Us vs. Them” culture. In the context of ServiceSoft, successful integration of UX work would require trust

between the two domains, discontinuation of defensive behavior, and a joint ownership of how a product must be developed. The situation called for concise user stories to establish constructive communication and to rebuild the trust between the two teams.

At ServiceSoft, verbose user stories were an indicator of deeply rooted collaborative and communicative issues. The DT distrusted the rigor of the PT's work and feared changes that would not create value; therefore, it questioned each user story. The PT referred to the verbose user stories instead of facing the "lions". Verbose user stories became a wall that limited mutual trust and knowledge sharing. Thus, verbose user stories sustained a vicious cycle of defensive behavior and erroneous assumptions regarding the user and the technical options.

To evaluate the agility of the practice at ServiceSoft, we used Conboy's three-point taxonomy of agility (Conboy, 2009). First, verbose user stories that were written weeks in advance and rarely changed impeded ServiceSoft's ability to react to change despite being in a fast-changing market environment. Second, it took long time to write the verbose user stories, which made the process of information sharing expensive. Third, because producing verbose user stories was time-consuming they were not continually ready to be utilized when needed because the process was ineffective and inefficient for the establishment of what had to be done and how. Thus, in the context of ServiceSoft, verbose user stories hindered the agility of software development. To address the identified challenges and create a change within ServiceSoft, we initiated our first intervention to improve integration of UX work with agile software development.

4.2. First intervention – Concise user stories

To initiate change, we started with the product manager because she wrote most of the user stories and held a position of power in the PT. The product manager is a key person. She initiates workshops, process changes and has every-day contact with both the PT and the DT. She is a mediator between the two domains. We chose to speak with a key stakeholder because it increases the success rate for any future interventions due to her position of power (Schaffer, 2004). The purpose of the intervention was to shift the product manager's perception of how user stories should be applied in the agile environment – as conversation starters rather than specifications.

4.2.1. Twenty more sentences

We initiated the intervention by presenting the problem situation and confirmed the issue of verbose user stories. Afterwards, we presented the possible solution – concise user stories. The argument was that concise user stories would encourage sharing of tacit knowledge, thus minimizing the risk of a product being developed based on the erroneous assumptions, because the PT would be able to invalidate them through conversation. Initially, the product manager's response to this approach was positive. She commented as follows:

Oh, that is interesting. Because I have gotten so many user stories back because guys say, "This is not specific enough, I don't know what this is". Okay fine, I will specify it with 20 more sentences for you.

[Product Manager - PT]

When coming across a communicative barrier (e.g., a member of the DT not understanding the full scope of a user story), the product manager chose to write, rather than communicate verbally. The overall process of information sharing was affected by this preference towards written communication. While the agile manifesto promotes the principle of face-to-face communication, UX work does not imply a particular method for conveying information.

4.2.2. I will challenge you no matter what you say

The product manager remained unconvinced that concise user stories were applicable in the context of ServiceSoft.

I see the point. And it sounds like a nice method. However, since the motivation for it is to make developers ask questions, or to inspire them to ask questions. However, instead of asking questions about the user, they actually question the whole user story, and that I don't think is knowledge sharing, that is more "You tell me this, but yet I don't believe you and I will challenge you no matter what you say".

[Product Manager - PT]

The product manager experienced the DT's distrust of the user stories as criticism towards her work rather than exploration. The product manager felt that she had to act cautiously to guard against unconstructive criticism from the DT. Thus, she used verbose user stories as a shield to protect herself and the stories from the DT's criticism, sustaining the defensive behavior that undermined the agility. She dismissed the idea of concise user stories, believing that "the more precise you get, the better results come out of it" (Product Manager - PT). The product manager experiences that verbose user stories lead to "fewer questions, meaning less resistance to actually take up the user story" (Product Manager - PT). The situation, which is described by the product manager, fits with Cohn's description of a poor practice. When a developer believes that a story card reflects all the details of a story, then "there's no further need to discuss the story" (Cohn, 2004, p. 20). However, the product manager associated successful information sharing - "better results" - with fewer questions and less argumentation.

While the product manager rejected our intervention, we still found her resistance to concise user stories enlightening. We found that preferences towards written communication and professional pride are also influencing the problem situation. Our conclusion was that we needed to involve both the PT and the DT because the problem situation required engagement from both teams – not only the product manager. Therefore, the second intervention addressed both teams to shift their current collaborative practice from verbose user stories towards concise user stories.

4.3. Second intervention – Deliberation on user stories

The second intervention at ServiceSoft involved both the DT and the PT in a workshop. Three PT members and five DT members participated. The workshop had the overall goal of initiating reflective conversation regarding the current problem situation with user stories and propose a solution.

4.3.1. In love with the product

We started out by presenting the problem situation; how the user story method is applied in the context of ServiceSoft, how it affects collaboration, and how it could be applied. Both the PT and the DT recognized that knowledge sharing was challenging because "it is difficult to be aware of things you know" (Product Expert - PT). Even the product manager was open to the proposed change towards more concise user stories, and expressed that she would like "to see whether it (read concise user story) could be a better solution". During the first intervention, the product manager was convinced that verbose user stories contributed to acceptance of a presented user story, while concise user stories lead to doubt and disbelief. Nevertheless, she expressed eagerness to try the recommended approach to concise user stories and reciprocal communication. The product manager highlights the issue of not being in love with the problem as a fundamental issue of UX integration in the ServiceSoft context:

We should fall in love with the problem (read user story) and not with the solution (read product). I think that we get too attached to the solution and almost try to falsify the problem. [W]e have destroyed many user stories by falsifying them.

[Product Manager - PT]

The product manager argued that the DT and the PT were “in love” with the product, which led to occasional invalidations of user stories. The PT and the DT were not only defensive towards each other but also towards the users. They undermined the severity of the problem and argued against the user’s needs in order to defend the product. Thus, the issue was not only between the PT and the DT, but also impacted the end user. Both the PT and the DT agreed that “there should not be such a thing as falsifying a problem”, and simply stating that “the users should know better” is not a solution (Product Manager - PT).

4.3.2. We are all in this together

The product manager emphasized that they had a common goal and shared the responsibility of reaching this goal together, as she noted:

We are just two teams. There are other teams in the organization, and they are all responsible for the user story, every single one of us. We all have the solution to the problem; it is both of those things. We are all in this together.

[Product Manager - PT]

The product manager stated that the entire ServiceSoft organization was responsible for a user story. Thus, integration of UX was not only a question of merging two conflicting domains; it was an organizational transformation.

4.3.3. The verbal part that needs to be there

After the presentation, we asked the PT and the DT to reflect on good experiences in their collaboration. Both the PT and the DT agreed that good collaboration implies “open-mindedness” and “keeping ego out of the equation”. We found, that in order to encourage a good collaborative practice, both the PT and the DT need to “find a solution together” (User On-boarding Specialist - PT), “understand each other” (Front-end Developer - DT), and be willing to “change one’s mind” (Back-end Developer - DT). The product manager noted that open-mindedness coincided with an ability to reach a common agreement on the basis of a conversation, and emphasized a need for an “environment where we all got a common agreement on what we are looking at” (Product Manager - PT). The PT and the DT’s willingness to accept each other’s differences and compromise was the way of best addressing the issue of the verbose user stories becoming a wall between them.

ServiceSoft was striving for a deliberative practice; however, they were unable to create the environment that supported deliberation. Therefore, through the workshop on the sensitive topic of collaboration, we established how deliberation could be supported in the context of ServiceSoft. An open and reflective conversation concerning the collaborative issues resulted in a common agreement regarding the actionable outcome of the workshop:

[Head of Development – DT]: The actionable outcome here, am I correct here that we are going to try to ignore or limit descriptions in user stories, to avoid biases and assumptions to see how that goes?

[Product Manager - PT]: Yes.

[Front-end Developer - DT]: And then, of course, the verbal part that needs to be there for us to be effective. So, make the descriptions more open.

[Conversation between the workshop participants]

The head of development presented the changes that would be implemented, both PT must “try to ignore or limit descriptions in user stories” in order “to avoid biases and assumptions to see how that goes”. The product manager was going to facilitate these changes because she wrote the majority of the user stories. Therefore, it was important that the product manager was engaged in the process of change, which she had expressed multiple times during the workshop by stating that she is “up for trying something else” (Product Manager - PT). The front-end developer highlighted the verbal communication as a tool which might assist an accomplishment of these changes. Thus, through the process of deliberation, we were able to initiate change, which might lead to an improved integration of UX work with agile software development in the context of ServiceSoft.

4.3.4. A workshop about how to have a workshop

Two months after the workshop, we did two follow-up interviews with the product manager and the head of development to evaluate how the change affected the integration of UX work from the standpoint of agility (Conboy, 2009). Both the product manager and the head of development stated that ServiceSoft was in the process of eliminating verbose user stories, but had not yet succeeded. During the workshop, the core issues had been vocalized, but they were not easily solved in their daily practice. Therefore, ServiceSoft focused on solving the underlying collaborative issues.

In the past two months people are more willing to stand up and show initiative and cooperate. We agree that maybe “my way is not the best way”. My own attitude has changed – the way I deliver the message. But I have not cracked it (read user story) yet. We have discussions, and are talking about how we should communicate. We had a workshop about how to have a workshop, and how to make the two teams work better together.

[Product Manager - PT]

Management now strives for emergence of trust between the two teams. They involve relevant stakeholders from both teams to negotiate how collaboration between the PT and the DT should be cultivated. Even though ServiceSoft was not able to completely eliminate verbose user stories, they adopted a deliberative practice towards making them more concise. Management has “created a space where big user stories can get a proper attention” (Head of Development - DT). Both the PT and the DT now “work together”, and “involve other major stakeholders, like our CEO or Head of CSM, if a user story in any way concerns them”. (Head of Development - DT). They recognized the value of deliberation and of using concise user stories; a concise user story was continually ready and easily changed. A concise user story allowed the PT to use less time on documenting information, and more time on disseminating the knowledge. However, even this small change towards concise user stories for integrating UX work with agile development was complicated and not easily achieved.

5. Discussion

Integrating UX work with agile development is a considerable challenge that has been a research interest for over a decade (Brhel et al., 2015; da Silva et al., 2018) with many studies of its challenges and success factors (da Silva et al., 2013; Brhel et al., 2015; da Silva et al., 2018; Kashfi et al., 2019). However, few studies are concerned with the problem identification and improvement in practice through AR. Against this backdrop, we present two key contributions from our two AR iterations at ServiceSoft. First, a user story, albeit a small part of a complex practice, can be an indicator of deeply rooted UX integration problems. Second, deliberation with concise user stories helps UX integration.

5.1. The user story as an indicator of integration

At ServiceSoft, UX professionals applied user stories to mediate user needs to the developers. However, the developers questioned the credibility of user stories delivered by a UX professional – rather than by a user. A problem of agile developers' distrusting UX professionals has been well described in previous research (Bruun et al., 2018; Chilana et al., 2010). At ServiceSoft, the UX professionals wrote extensive context descriptions in the user stories and avoided verbal communication, which directed developers' attention to semantics and sentence structures. This practice with user stories sustained a gap between the two professions and impaired their agility, according to our evaluation using Conboy's (2009) framework. Extensive context descriptions written weeks prior to implementation hindered their reaction to change, thus disregarding the fast-moving environment of agile software development (Persson et al., 2016). Moreover, the UX professionals providing solutions to a user story, which developers treated as definitive without exploring further possibilities, impaired the creation of change. Thus, user stories with extensive context descriptions – verbose user stories – indicated unsuccessful UX integration and limited agility. This finding nuances the current assumption that artifact mediated communication is an element of a successful UX integration (Brhel et al., 2015; Andrei et al., 2017; Kashfi et al., 2019).

We identified properties of a user story that can negatively affect integration of UX, e.g. verbose and solution oriented, or positively affect integration of UX, e.g. concise and ready to change. However, concise user stories do not in themselves indicate successful integration of UX with agile development. Schmitz et al. (2018) argued that *terse* user stories and subsequent discussions do not further common understanding of user needs. They stated that “the communication methods used in Blue Velvet allowed the developers to remain mentally entangled in their own vision for the system” (p. 38). On the basis of this, we argue that researchers ought to understand an artifact in situ: recognize how practitioners use an artifact and why, identify stakeholders involved and how they are affected by an artifact in use.

We present a more nuanced view on the type of communication required in order to achieve a common vision for the system. Concise user stories *need* deliberation.

5.2. Deliberation for integration

At ServiceSoft, verbose user stories hindered deliberation and lead to a “mistaken belief that the story cards reflect the details and that there is no further need to discuss the story” (Cohn, 2004, p. 20). Deliberation concerns “mutual communication that involves weighing and reflecting on preferences, values and interests regarding matters of common concern” (Mansbridge, 2015, p. 27). “Mutual” implies a two-way communication; “weighing and reflecting” regard rational and thoughtful consideration; and lastly, “preferences, values and interests on matters of common concern” refer to the importance of a collective and not an individual (Mansbridge, 2015, p. 29). Deliberation signifies the importance of a thoughtful, reasonable and reflective practice, with a purpose of achieving a common understanding through argumentation and consideration of different claims (Marques and Maia, 2010; Mansbridge, 2015). Decisions should be based on fair and reasonable discussions, while sharing and building knowledge as a group (Marques and Maia, 2010). Thus, UX work is not established exclusively by a UX professional, it is built through a deliberative practice with agile software developers. This insight complements findings presented by Kashfi et al. (2019). They advocate a shared ownership over UX work as a way to improve collaboration between UX and non-UX practitioners (Kashfi

et al., 2019). Jones and Thoma (2019) found that lack of shared decision-making, and limited autonomy hinder UX integration. They recommend close proximity, early and frequent communication, shared ideation and problem solving, crossing over of knowledge and skills, co-creation and prototyping and making joint decisions (Jones and Thoma, 2019), which complements the notion of deliberation.

In ServiceSoft's integration of UX with agile development, concise user stories imply deliberation. Nevertheless, the concise user stories are also a product of deliberation. Dourish (2017) described the interdependence between the deliberation and an artifact. He stated that an *editable* artifact, which user stories are, focuses attention on the idea that what is going on is not simply communication but deliberation, and that the document that anchors the meeting is one to be not simply produced but transformed within the context of the meeting itself” (Dourish, 2017, p. 101). Dourish (2017) view supports our finding that an artifact enforces deliberation and, at the same time, is formed by it.

At ServiceSoft, the UX professionals were not able to apply concise user stories due to the lack of deliberation, which led to uncertainty and defensive behavior that was manifested in extensive context descriptions.

Chilana et al. (2010) present the challenge of distrust towards UX work, and how it negatively affects UX practice and collaboration between stakeholders. Chilana et al. (2010) argue that in order to improve the UX work the long-term educational changes are needed. We argue that through deliberation, UX professionals and agile developers can agree on a possible solution to a common concern – a user story. Deliberation can reduce the developers' distrust in a user story and in UX professionals' work, as well as reaffirming UX professionals' trust in the developers and allowing them to dismiss the need for extensive context description. Thus, deliberation aids an emergence of concise user stories, which becomes an interconnected practice for successful integration of UX with agile development. However, the deliberative practice required a common action (Ferreira et al., 2011), both teams needed a change in how they perceive themselves and each other (Section 4.1). Through “knowledge-building discursive processes” (Marques and Maia, 2010, p. 615), they can come to acknowledge each other as an integrated part of one whole. However, the interconnectedness of the described practice does not allow a stepwise guide to achieving such desired change. Deliberative theory states that “when talking and exchanging points of view, individuals may improve thought patterns and the interpretation of ... issues” (Marques and Maia, 2010, p. 615). Yet, our study showed that addressing the issue through a conversation with a *single* person, a key UX professional in an influential position, did not initiate the change. Only through deliberation – open, inclusive, reflective, reciprocal discussion with relevant stakeholders – were we able to improve their integration. Thus, we propose that deliberative practice is not only a solution to a successful UX integration, but also a means to initiate change towards successful UX integration.

5.3. Transformation in integration

Integrating UX work with agile software development is a transformative process. Just as Stolterman and Fors (2004) understand digital transformation as “the changes that the digital technology causes or influences in all aspects of human life” (p. 689), we understand a transformation in integration as the changes in practice that affect all aspects of software development. The UX professionals transform user experiences into artifacts (Brhel et al., 2015) that the agile software developers transform “into working software” (Ferreira et al., 2011, p. 967).

Their collaborative practice is transformed through deliberation on the artifacts they use among themselves (Dourish, 2017). This transformation is a knowledge-building discursive process towards organizational change (Kashfi et al., 2019), which is not easily achieved (Schein, 1996).

Existing literature on UX integration with agile software development often focuses on large transformations, such as establishing shared practices (Ferreira et al., 2011), processes (Sy, 2007), and methods (Andrej et al., 2017). We argue that in order to achieve a transformation in integration on a larger scale, we ought to first focus on the small transformations. Schein (1996) states that the key to an effective change is “the ability to balance the amount of threat” (p. 30), which means that the change should not be overwhelming for an individual or a group. Thus, large-scale transformation in integration can hinder successful and sustainable change in practice. We as researchers need to understand the small transformations in practice, how they occur, and why, which requires the high level of involvement afforded by action research.

6. Limitations

Our study focused on how user stories and deliberation affect collaboration on the activities prior to software implementation in exploring and establishing a common understanding of user needs and technological limitations. The user’s perspective is a continuous concern; therefore, future work is needed to understand how concise user stories and deliberation affect other activities in agile software development. Furthermore, in follow-up interviews with the product manager and the head of development at ServiceSoft, we found that they continued to recognize the value of concise user stories and deliberation but struggled to reinforce it. Therefore, further studies are needed to examine reinforcement practices for the integration of UX work with agile software development on different organizational levels. This is a concern that is similar to improving agility on different levels in a fast-moving software organization (Persson et al., 2016).

Another limitation of our study concerns our research method. AR concerns “local solutions to local problems” (Hayes, 2011, p. 16), where the researchers and the community partners collaborate on finding relevant solutions to real problems. This research process limits control of the study (Kashfi et al., 2019; Mckay and Marshall, 2001) and does not promote the generalizability and neutrality usually associated with scientific rigor (Mckay and Marshall, 2001; Hayes, 2011). Some researchers might find AR lacking in the scientific merit, question the validity of AR and the subjectivity of the researcher (Avison et al., 2018). The validity of AR is expressed through *trustworthiness* that embodies credibility, transferability, dependability, and confirmability (Hayes, 2011, p. 7). Credibility of our study stems from a high level of involvement, which enabled us to obtain tacit knowledge and firsthand experience, which are difficult to gain through interviews and observations. Our findings might be transferable; therefore, further research is required in order to establish transferability of our findings on concise user stories and deliberation in similar contexts. Dependability was ensured through biweekly meetings that the first author had with the second and third authors. The second and third authors were independent entities throughout the study, ensuring inquiry audit. Furthermore, the collaborative and democratic nature of AR enabled us to confirm our findings, which ensured that both the solution and the problem were relevant in the ServiceSoft context. Thus, potential biases of the first author were elicited by the second and the third author, and the company. We were aware of the biases and we embraced them. AR requires researchers not to perceive the interventions and

their presence “as “contamination” or “bias” but as an inevitable part of the social construction of scientific knowledge” (Hayes, 2011, p. 7).

Kashfi et al. (2019) criticized the concept of *intervention* in AR for limiting the researcher to only investigate “the impact of these manipulations” (p. 39). They argued that action researchers disregard “other types of events that in a real industrial setting may influence integration over time” (p. 39). However, AR holds that “The best way to understand something is to try to change it” (Kurt Lewin as cited in Hayes, 2011). In our real industrial setting, we were forced to consider a wide range of events to understand what needed to be changed and why. However, we recognized that, after the problem situation had been established, we focused on the importance of user stories and may have disregarded other possible events important to the integration of UX work with agile software development.

7. Conclusion

Integrating UX work with agile software development is complex, contextualized, and difficult in practice. From two AR iterations in a small software company, we showed how concise user stories and deliberation are useful focuses. Verbose user stories with voluminous context descriptions are artifacts that, as such, should support collaboration between UX professionals and agile developers (Garcia et al., 2019). However, our study showed that verbose user stories became a wall between the two practices and contributed to sustained defensive behavior. Each verbose user story became another brick in the wall, further segregating the two practices with erroneous assumptions regarding the user and technology. We propose based on our AR study that deliberation involving inclusive, reflective, and reciprocal communication based on concise user stories help improve the integration of UX work with agile software development.

CRedit authorship contribution statement

Alisa Ananjeva: Conceptualization, Data curation, Formal analysis, Investigation, Writing - original draft. **John Stouby Persson:** Methodology, Conceptualization, Supervision - ensuring that data curation is focused, Writing - review & editing. **Anders Bruun:** Conceptualization, Supervision - ensuring that data curation is focused, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Andrej, G., Tiago Silva, d.S., Milene, S.S., 2017. Artifacts for agile user-centered design: a systematic mapping. In: HICSS.
- Avison, D., Davison, R., Malaurent, J., 2018. Information systems action research: Debunking myths and overcoming barriers. *Inf. Manage.* 55 (2), 177–187.
- Avison, D.E., Lau, F., Myers, M.D., Nielsen, P.A., 1999. Action research. *ACM* 42 (1), 94–97.
- Barksdale, J., McCrickard, D., 2012. Software product innovation in agile usability teams: an analytical framework of social capital, network governance, and usability knowledge management. *Int. J. Agile Extreme Softw. Dev.* 1, 52–77.
- 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.
- Brinkmann, S., 2014. Doing without data. *Qual. Inq.* 20 (6), 720–725.
- Brown, J., Lindgaard, G., Biddle, R., 2011. Collaborative events and shared artefacts: Agile interaction designers and developers working toward common aims. In: *Proceedings - 2011 Agile Conference, Agile 2011*, pp. 87–96.

- Bruun, A., Larusdottir, M.K., Nielsen, L., Nielsen, P.A., Persson, J.S., 2018. The role of UX professionals in agile development: A case study from industry. In: *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*. NordiCHI '18, ACM, New York, NY, USA, pp. 352–363.
- Bruun, A., Stage, J., 2015. New approaches to usability evaluation in software development: Barefoot and crowdsourcing. *J. Syst. Softw.* 105, 40–53.
- Chilana, P.K., Wobbrock, J.O., Ko, A.J., 2010. Understanding usability practices in complex domains. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10, ACM, New York, NY, USA, pp. 2337–2346.
- Cohn, M., 2004. *User Stories Applied: For Agile Software Development*, first ed. Addison-Wesley, pp. 0–29.
- Conboy, K., 2009. Agility from first principles: Reconstructing the concept of agility in information systems development. *Inf. Syst. Res.* 20 (3), 329–354.
- da Silva, T.S., Silveira, M., Maurer, F., 2013. Ten lessons learned from integrating interaction design and agile development. In: *Proceedings - AGILE 2013*, pp. 42–49.
- da Silva, T.S., Silveira, M.S., Maurer, F., Silveira, F.F., 2018. The evolution of agile UXD. *Inf. Syst. Softw. Technol.* 102, 1–5.
- da Silva, T.S., Silveira, F.F., Silveira, M.S., Hellmann, T., Maurer, F., 2015. A systematic mapping on agile UCD across the major agile and hci conferences. In: *Computational Science and Its Applications - ICCSA 2015*. Springer International Publishing, pp. 86–100.
- Detweiler, M., 2007. Managing UCD within agile projects. *Interactions* 14, 40–42.
- Dourish, P., 2017. *The Stuff of Bits: An Essay on the Materialities of Information*, first ed. The MIT Press.
- 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.
- Fitzgerald, B., 1997. The use of systems development methodologies in practice: A field study. *Inf. Syst. J.* 7, 201–212.
- Garcia, A., da Silva, T.S., Silveira, M.S., 2019. Artifact-facilitated communication in agile user-centered design. In: *Agile Processes in Software Engineering and Extreme Programming*. Springer International Publishing, pp. 102–118.
- Hayes, G., 2011. The relationship of action research to human-computer interaction. *ACM Trans. Comput.-Hum. Interact.* 18, 15.
- Jones, A., Thoma, V., 2019. Determinants for successful agile collaboration between UX designers and software developers in a complex organisation. *Int. J. Hum.-Comput. Interact.* 35 (20), 1914–1935.
- Kashfi, P., Feldt, R., Nilsson, A., 2019. Integrating UX principles and practices into software development organizations: A case study of influencing events. *J. Syst. Softw.* 154, 37–58.
- Kuusinen, K., Mikkonen, T., Pakarinen, S., 2012. Agile user experience development in a large software organization: good expertise but limited impact. In: *Hum.-Center. Softw. Eng.*. Springer Berlin Heidelberg, pp. 94–111.
- Larusdottir, M., Gulliksen, J., Åsa Cajander, 2017. A license to kill – improving UCSD in agile development. *J. Syst. Softw.* 123, 214–222.
- Lazar, J., Feng, J., Hochheiser, H., 2017. *Research Methods in Human-Computer Interaction*, second ed. Elsevier Science & Technology, pp. 1–560.
- Lucassen, G., Dalpiaz, F., van der Werf, J.M.E.M., Brinkkemper, S., 2015. Forging high-quality user stories: Towards a discipline for agile requirements. In: *2015 IEEE 23rd International Requirements Engineering Conference*. RE, pp. 126–135.
- Lucassen, G., Dalpiaz, F., van der Werf, J.M.E.M., Brinkkemper, S., 2016. The use and effectiveness of user stories in practice. In: *Requirements Engineering: Foundation for Software Quality*. Springer International Publishing, pp. 205–222.
- Mansbridge, J., 2015. A minimalist definition of deliberation. In: *Deliberation and Development: Rethinking the Role of Voice and Collective Action in Unequal Societies*. World Bank Group, pp. 27–50 (Chapter 2).
- Marques, A., Maia, R., 2010. Everyday conversation in the deliberative process: An analysis of communicative exchanges in discussion groups and their contributions to civic and political socialization. *J. Commun.* 60, 611–635.
- Mckay, J., Marshall, P., 2001. The dual imperatives of action research. *Inf. Technol. People* 14, 46–59.
- McLellan, E., MacQueen, K.M., Neidig, J.L., 2003. Beyond the qualitative interview: Data preparation and transcription. *Field Methods* 15 (1), 63–84.
- Melnik, G., Maurer, F., 2004. Direct verbal communication as a catalyst of agile knowledge sharing. In: *Proceedings of the Agile Development Conference, ADC 2004*, pp. 21–31.
- Myers, M., Newman, M., 2007. The qualitative interview in IS research: Examining the craft. *Inf. Organ.* 17, 2–26.
- Nielsen, P., Persson, J., 2016. Engaged problem formulation in IS research. *Commun. Assoc. Inf. Syst.* 38, 720–737.
- Paay, J., Sterling, L., Vetere, F., Howard, S., Boettcher, A., 2009. Engineering the social: The role of shared artifacts. *Int. J. Hum.-Comput. Stud.* 67, 437–454.
- Peres, A.L., da Silva, T.S., Silva, F.S., Soares, F.S.F., de Carvalho, C.R.M., de Lemos Meira, S.R., 2014. AGILEUX model: Towards a reference model on integrating UX in developing software using agile methodologies. In: *2014 Agile Conference*, pp. 61–63.
- Persson, J., Nielsen, P., Bruun, A., 2018. Agility for UX and development: A case study. In: *10th Nordic Conference on Human-Computer Interaction*. In: NordiCHI '18, Workshop Position Paper.
- Persson, J., Nørberg, J., Nielsen, P., 2016. Improving ISD agility in fast-moving software organizations. In: *ECIS 2016 Proceedings*. The Association for Information Systems (AIS).
- Schaffer, E., 2004. *Institutionalization of Usability: A Step-by-step Guide*, second ed. Addison-Wesley Professional, pp. 37–47.
- Schein, E.H., 1996. Kurt Lewin's change theory in the field and in the classroom: Notes toward a model of managed learning. *Syst. Pract.* 9 (1), 27–47.
- Schmitz, K., Mahapatra, R., Nerur, S., 2018. User engagement in the era of hybrid agile methodology. *IEEE Softw.* PP, 1.
- Schwaber, K., Beedle, M., 2002. *Agile Software Development with Scrum*. Prentice Hall, Upper Saddle River, New Jersey.
- Spall, S., 1998. Peer debriefing in qualitative research: Emerging operational models. *Qual. Inq.* 4, 280–292.
- Stolterman, E., Fors, A.C., 2004. Information technology and the good life. *Inf. Syst. Res.* 687–692.
- Sy, D., 2007. Adapting usability investigations for agile user-centered design. *J. Usability Stud.* 2 (3), 112–132.
- Wale-Kolade, A., Nielsen, P., Päiväranta, T., 2014. Integrating usability practices into agile development: a case study. In: *ISD*.
- Zaitsev, A., Gal, U., Tan, B., 2020. Coordination artifacts in agile software development. *Inf. Organ.* 30 (2), 100288.

Alisa Ananjeva, M.Sc., is a Ph.D. fellow at the Department of Computer Science at Aalborg University, Denmark. Alisa's research interest lies in the interface between Information Systems (IS) and Human-Computer Interaction (HCI). She is interested in problem-based research and research methodologies such as Action Research and Action Design Research. Alisa aims at finding and solving problems in collaboration with the industry practitioners.

John Stouby Persson, Ph.D., is an associate professor at the Department of Computer Science at Aalborg University, Denmark. He does collaborative practice research on managing the development of information systems. Following a pragmatist philosophy, his research interests pertain to the management of global software development, agility, digital transformation, business cases, and benefits realization. In this context, he has published qualitative studies and action research of agile software development over the past 10 years in predominantly Information Systems research outlets.

Anders Bruun, Ph.D., is an associate professor at the Department of Computer Science, Aalborg University, Denmark. Anders' research in HCI deals with theoretical, methodological and practical aspects of user experience and interaction design. This research emphasizes development and assessment of novel methods that reliably measure UX in digital products, including a combination of quantitative real time measurements from wearable sensors as well as post-hoc qualitative responses from users. Anders' research interests also include studies of the challenges arising in the intersection between software development and interaction design practices. Industry collaborations have led to publications on e.g. identifying and overcoming challenges related to integrating user experience and software development practices.