Agile User Experience Development in a Large Software Organization: Good Expertise but Limited Impact

Kati Kuusinen, Tommi Mikkonen, and Santtu Pakarinen

Tampere University of Technology, Tampere Finland
{kati.kuusinen, tommi.mikkonen, santtu.pakarinen}@tut.fi

Abstract. While Agile methods were originally introduced for small, tightly coupled teams, leaner ways of working are becoming a practical method to run entire enterprises. As the emphasis of user experience work has inherently been on the early phases before starting the development, it also needs to be adapted to the Agile way of working. To improve the current practices in Agile user experience work, we determined the present state of a multi-continental software development organization that already had a functioning user experience team. In this paper, we describe the most prevalent issues regarding the interaction of user experience design and software development activities, and suggest improvements to fix those. Most of the observed problems were related to communication issues and to the service mode of the user experience team. The user experience team was operating between management and development organizations trying to adapt to the dissimilar practices of both the disciplines.

1. Introduction

User experience (UX) plays a significant role in the success or failure of contemporary software-centric products and services, as well as the companies producing them [15]. The fundamental goal of UX work is to create software that is highly usable and fulfills user needs. UX, standardized as a "person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service," is inherently based on the iterative model of human-centered design (HCD) [16], where context of use and user requirements are specified, and software is developed to meet those requirements. Yet, most prevailing Agile software development methodologies do not give clear guidance on how to incorporate user experience design activities in the software engineering practices. For instance, the Agile Manifesto [1] itself ignores UX-related activities.

In general, Agile development [1] refers to using iterative and incremental approach that emphasizes collaboration, feedback and working software over

processes, documentation and strict plans. By Agile UX we refer to work that systematically results in desired user experience of the outcome and is conducted according to Agile principles.

This paper describes the state-of-the-practice of Agile UX work in a large software organization that provides specialized software systems, mainly for internet service provider (ISP) markets. The company, one of the front-runners in the Agile transformation on the global scale, has several sites on various continents, and these all work together in an Agile manner. While their software development is aligned with Agile practices, the company has had problems in integrating UX work with Agile development. Our aim is to improve the current situation in Agile UX work within the company, and more generally to reveal good practices in Agile UX. Moreover, the goal is to uncover impediments and issues (both organizational and methodological) that support Agile UX inside the company. The study consisted of a web survey with 50 questions and 76 respondents which was followed by 13 face-to-face semi-structured interviews. We also studied internal documentation and compared it with the results of the interviews and the survey.

The rest of the paper is structured as follows. Next, in Section 2, we provide some related work and background on user experience and Agile development. In Section 3, we introduce the ways of working in the organization in which the study has been carried out. In Section 4, we discuss how the actual study was implemented, and in Section 5, we list the main lessons we have learned from the study. In Section 6, we give an extended discussion of the study and possible future actions. Towards the end of the paper, in Section 7, we draw some final conclusions.

2. Background and Related Work

Previous research has indicated there is a clear need for further studies of UX work in the field of Agile development [12, 25]. Silva da Silva et al. [25] conducted extensive literature research regarding HCD (including UX) and Agile methods. They presented 58 research articles found relevant in their study, and pointed out that most studies have been descriptions and conclusions of experiments. Therefore, they conclude that further studies are needed to understand the best practices in Agile UX work.

2.1 Practicing Agile UX Work

There is no clear consensus on best practices in Agile UX work – in other words, UX work that is conducted in accordance with Agile principles and methods, thus integrating (or merging) UX work and HCD practices with Agile development practices. There is clear evidence that some or little design upfront (SDUF or LDUF, respectively) is needed also with an Agile approach; this is recommended in 31 papers analyzed by Silva da Silva et al. However, the reason those papers recommended LDUF is that big design upfront is against Agile principles, and these articles present no recommendations on which design practices should be used during design upfront. Besides the need for SDUF, another evident finding is that integrating UX design and

other development activities improves communication and collaboration between the functions; this is reported in 26 articles presented by Silva da Silva et al. Common practices in Agile HCD work include low fidelity prototypes, user tests, and user stories. However, ways of utilizing these practices vary. User testing on paper prototypes and on working software were equally recommended [25].

Common problems in Agile UX work include power struggles, differences in schedules between HCD and implementation, communication issues, unwillingness to understand project needs, and failure to achieve the right amount of user involvement [6]. Budwig et al. [4] report problems in understanding the big picture. UX work conducted within a development sprint caused confusion and problems in communication and off-shore coordination [4]. Petrovic et al. [22] state that in many cases, UX specialists are those who end up in quality assurance work. The above issues still remain unsolved, since they are considered as too expensive or difficult to improve. In addition, design vision and information architecture ought to be understood before starting implementation, and their realization assured in development. Such practices prevent the situation in which problems are discovered only after implementation [22].

Ferreira et al. [11] observed that upfront design improves user satisfaction and product consistency. Upfront design assisted in finding affordable design solutions. It also supported project-level estimation and prioritization, and thus led to savings in time and costs. Federoff and Courage [10] report that Agile UX work has been improved by parallel development and design, working one sprint ahead, interactive prototypes for communicating design, and design studios. Budwig et al. [4] report that a UX team benefitted from close collaboration with a broad cross-functional team since issues were found earlier and addressed faster. They also recommend that the UX team should work one to two sprints ahead of development teams in their own sprints, and should be co-located with development teams. The UX team should also work with business people to define requirements before starting UX work. Moreover, the UX team should already be involved in roadmap work. Working closely together helps in sharing information, and starting UX work early enables recognizing and considering user expectations in time, that is, when UX still is affordable and leads to optimal savings in development time and costs.

Chow and Cao [7] conducted multiple regression analysis on 48 common hypotheses of success factors in Agile software development and determined that only 10 were supported. From those, the authors found that the truly critical success factors were these: correct delivery strategy, proper practice of Agile techniques, and a competent team. The second set of important factors included good Agile project management process, a cooperative orally communicating team, and strong customer involvement.

Currently, one of the most recommended process models for applying UX work in an Agile context seems to be the "one sprint ahead" approach (Figure 1) originally proposed by Miller [21] and Sy [27], later applied or modified by Fox et al. [13], Silva da Silva et al. [25], and many others.
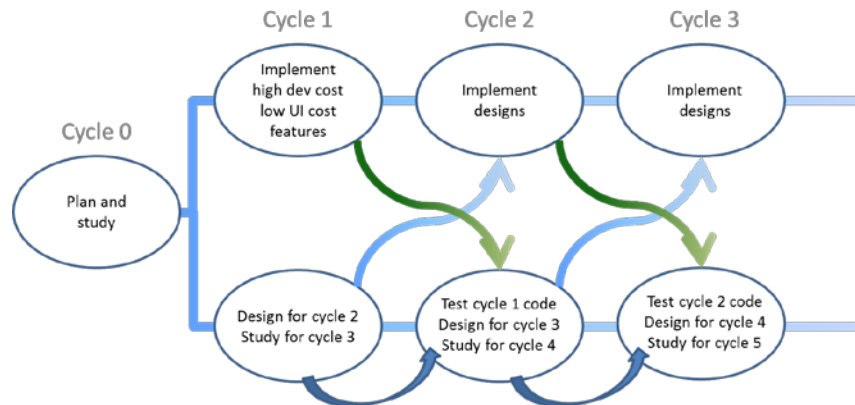
Figure 1. One sprint ahead approach. Redrawn from [27]

The process model of Miller [21] and Sy [27] has been in use at Autodesk, but no recent experience reports were available. One sprint ahead is also recommended by Budwig [4] and Federoff and Courage [10]. Indeed, as by definition, UX design is iterative [16], working at least one sprint ahead gives an opportunity for iterations. Sy [27] observed the approach helped to complete the design just in time, which resulted in less design waste. The approach helped to gain a shared vision; through daily contact developers could be aware of design progression and give their input early enough [21]. Miller found the close cooperation and daily interaction between developers and UX designers to be an essential success factor. She reported savings in design and development time and improvements in user satisfaction [21].

3. Way of Working in the Organization

In the following, we introduce the ways of working within the software development organization we have studied. First, we discuss research and development, where Agile practices play a major role. Then, we address certain other functions of the company, where other practices exist.

3.1 Research and Development

As the company being investigated is one of those that have been spearheading the Agile transformation, it is only expected that Scrum [24, 26] is extensively used in research and development. Scrum (Figure 2) is a simple, iterative framework for project management. When using Scrum, incoming requirements to be implemented are stored in Product Backlog (PB). They are implemented in terms of Sprints – iterations of a fixed length of two to four weeks. For each Sprint, a collection of Product Backlog items are selected and refined into Sprint Tasks. Then, these are implemented in tasks of the Sprint. After each Sprint, a complete system is available, that can be delivered to clients, at least in principle.

Only three roles are defined in Scrum: Product Owner (PO) is responsible for managing the Product Backlog; Team, consisting of developers, is responsible for executing the Sprint; and Scrum Master will eliminate any emerging impediment and is responsible for enforcing the Scrum process. At the heart of the Scrum process are the self-organized Scrum Team and committed involvement of PO. In addition, team stability is of crucial importance, as it is the Team that takes care of the implementation tasks as a single, high-performing and effective entity.

The company being studied has been following Scrum for six years. Due to the size of the company, they also utilize Scrum-of-Scrums, which coordinates the effort of all Scrum teams towards a single goal that is more comprehensive than that of any of the Scrum teams themselves.
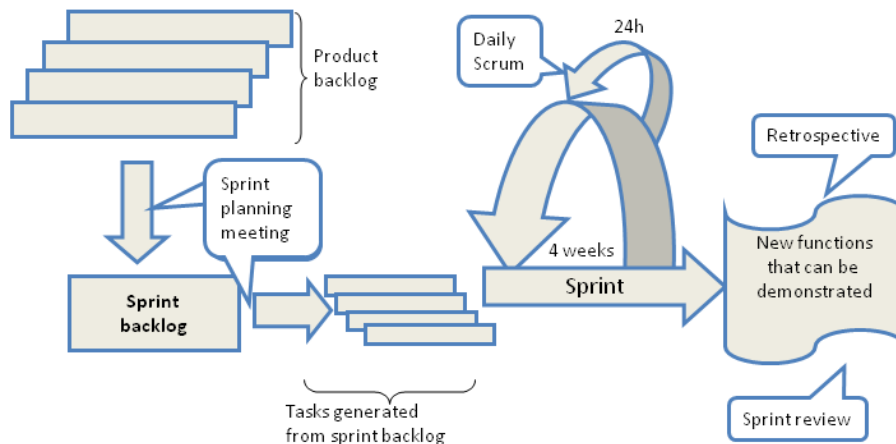

Figure 2. Scrum process

3.2 User Experience

The company has an established UX team with about 15 UX professionals. The team has existed for four years. The majority of the team is physically co-located in Finland, but some of the team is located in Asia. Organizationally, the majority of the UX team is allocated to product management, which is responsible for developing a concept of software before starting a project. Hence, UX specialists are usually involved in development early on. The UX team members have specialized roles. One is specialized in research activities, another is responsible for official communication inside the organization, and some are interaction or graphic designers. The UX team rarely works together as a team. Instead, members have certain projects or tasks they are working with as individuals or sometimes as pairs.

Fixed quarter and yearly plans structure product management's work whereas R&D is following Scrum in their work. In general, product management is responsible of defining the product and making early concepts. R&D implements the products. As the UX team members operate between R&D and product management, their ways of working have characteristics of both product management and R&D – in general, the

UX team's iterative practices do not follow Scrum. While the UX team members usually have some design upfront time, in some cases they start design work as implementation sprints begin. Generally, they adapt the practices of the function or team with which they are cooperating with.

Generally, UX issues are not in PB as such, and there are no acceptance criteria for UX work. However, UX team members working in projects follow PB. Usually developers get the design outside PB as wireframe or high-fidelity images. The UX team has earlier tried Agile working methods inside the team, too, but they were unable to stay one sprint ahead of development. Therefore, the trial was cancelled.

3.3 Other Parts of the Organization

Inside the organization, the use of Scrum is limited to R&D. In contrast, when delivering complete system-level solutions, for example, commonly composed out of the parts that are developed using Scrum, the process resembles a more conservative iterative approach, where releases are planned in advance according to the needs of the most important customers. The approach is iterative but it has fixed scope and time frame. In addition, there are certain management activities that do not follow Agile principles. Such topics fall beyond the scope of this paper.

4. Research Method

The explorative case study [18, 23] described in this paper was conducted by two researchers from a university research organization over a ten-month period from February to October 2011. The main research question was 1. How can the present state of Agile UX work be improved in the case organization. Other research questions were 2. What are the significant challenges in Agile UX work in the organization and how those can be resolved, and 3. What are the current good practices in Agile UX work. Since the case study was explorative, we did not have a hypothesis. We are also studying other companies with similar approach, e.g. [17].

4.1 Main Themes of the Case Study

The present state analysis was conducted in two parts: In the first part, a survey concentrated on collaboration and interaction, processes and tools, and respondents' knowledge about UX and usability; in the second part, a series of interviews concentrated on processes, collaboration, and acquiring and using user feedback. Processes and tools included both official and unofficial practicalities and working methods in the organization, such as the official development process and how employees follow it in practice. The collaboration and interaction section concentrated on how employees communicate with each other, which collaboration methods they would prefer, and what is hindering their cooperation. Knowledge was measured by

how respondents defined UX and usability and how they would rate their own expertise.

4.2 Survey

The web survey was designed to reveal the present state of UX work in Agile organizations in general (in this paper we focus on one particular organization). It consisted of 50 questions (31 open and 19 closed-ended) on collaboration and interaction, processes and tools, and concepts and knowledge. The questions were generated iteratively by two researchers, based on a variety of studies [3, 5, 7, 9, 14, 20, 28, 29, 31]. References were studied to determine significant areas and common problems to generate appropriate questions. Questions and concerns were collected from the mentioned studies and iterated into a survey by the two researchers. The survey was piloted in two companies and iterated based on the feedback before starting the data gathering for the study presented in this paper.

The survey was open for answers for three weeks in February 2011. Invitations to participate in the survey were sent to members of the R&D organization and to its interfaces, such as marketing and management, including the UX team. Interviews were conducted two months later, after the survey had been analyzed. Interview questions were generated to go deeper in the focus areas, and from topics that remained unclear upon analyzing the survey responses.

Open-ended questions of the survey were analyzed by two researchers with a qualitative, theory-bound analysis method with an emergent coding approach [18] as follows. First the researchers separately read the material and classified the data per question. After that the researchers discussed the categories and formed classes. Then the data were coded into classes individually by two subjective coders to quantify them. An inter-rater reliability analysis using the Cohen's kappa statistic was performed to determine the consistency among coders. Quantified data were analyzed with SPSS to find statistical significances. Theory was bounded to data in the analysis phase when linking findings with Agile and UX methodologies.

4.3 Interviews

Semi-structured interviews were conducted as individual or pair interview by one or two interviewers. About 70 to 80 minute interviews were recorded and analyzed from written transcripts. Interview questions covered the job content and tasks of each interviewee: which tasks are their responsibility, who they are working with, and how they communicate with UX specialists and users of their work output. Additionally, questions about utilized Scrum roles and practices were asked.

Interviewees were selected to cover the most focal roles in software development. The company contact person helped in arranging the interviews. Questions were grouped based on the respondents' role. Some of the questions were aimed at all the respondents, but each role also had role-specific questions.

The interviews were analyzed by using the affinity wall method as instructed by Beyer and Holtzblatt [2]. The data-driven method utilizes theory-independent coding on a physical wall. Transcribed interview data were worked into 1126 notes each containing one piece of information. Three researchers (two subjective and one objective coder) categorized notes either including one into an existing category or creating a new one if no suitable category was available. If the note was considered irrelevant to the study, it was excluded from the analysis. Later we labeled and revisited the existing categories, and grouped those under new upper-level categories.

5. Findings

Over the course of the study, it quickly became apparent that although the original goal was to study the relation between Agile development and UX design, the findings would also reveal numerous other organizational aspects. In the following, we introduce the sample and our main findings organized into four categories (understanding UX, UX team, and cooperation). While listed separately, the issues are often intertwined with each other, and improving the situation requires considering them all.

5.1 Survey Participants and Interviewees

Table 1 describes the job roles of the 76 survey respondents. They had been working for the company 2 to 16 years, mean 7 years. Total work experience varied between 4 and 24 years, mean 14 years. Thirty-three respondents (developers, architects, scrum masters, and POs) were from R&D, whereas the rest (38) were from other functions such as management and UX. We calculated Cohen's kappa for coding open-ended answers. The inter-rater agreement was found to be kappa = 0.885 with $p < 0.001$ (almost perfect).

Table 1. Survey respondents' job categories (open-ended question)

Job	N	Job	N
Developer	20	UX specialist	6
Management	19	Scrum master	4
Product manager	7	Product owner	3
Architect	6	No answer	5
Quality assurance	6		

Table 2 presents office locations of the respondents. All the interviewees and 56.9% of the survey respondents work at the head office in Helsinki, Finland, and thus the core of the findings is strongly linked with that site. Different sites have their own ways of working; there is no general process that all the sites would follow.

Table 2. Office locations of survey respondents (closed-ended question)

Office location	N	Office location	N
Helsinki	37	Kuala Lumpur	7
Bordeaux	9	St Petersburg	3
Oulu	7	Paris	2

Altogether 13 persons with different roles in development were interviewed. Table 3 presents interviewee roles and codes that are used in this paper when presenting results to reveal which of the interviewees are behind each finding. The interviewees had total work experience from 9 to 31 years, mean 15.1 years. They had been working for the company from 3 months to 17 years, mean 5.2 years. And they had working experience of Scrum from 2 to 5.5 years, mean 3.4 years.

Table 3. Interviewees' work roles and codes used in this paper

Code	Role	Code	Role
H1, H2	Product owner	H8, H9	Scrum master
H3, H4	UX designer	H10	Quality engineer
H5, H6	Architect	H12	UX manager
H7	Developer	H11, H13	Product manager

5.2 Overview of the Results

The survey revealed that problems in Agile UX work were mostly process and communication related. The expertise both in Scrum and UX was generally good. Of the survey respondents, 85.7% evaluated their expertise in UX on a six-level scale from inadequate (1) to adequate (6) to successfully complete their assignments four or higher, N=70. On a six-level scale from novice (1) to expert (6), 63.0% evaluated their expertise in UX four or higher. Scrum masters (75.0%), quality engineers (50.0%), product managers (42.8%), and managers (38.9%) considered their expertise in UX low most often. The biggest problems in Agile UX work were considered to be lack of cooperation between UX specialists and developers, lack of time for designing UX / getting UX designs for implementation too late, and balancing the amount of early concept creation work for understanding the big picture before starting a development project.

Interviews concentrated on the process, communication, and acquiring and utilizing user information and feedback. Both the survey and interviews indicated problems in communication due to lack of face-to-face time. In interviews we discussed through the development process and found improvable issues from every phase, such as lack of common practice for early UX work and verifying the implementation of UX design. It was also hoped that UX work should be more agile.

5.3 Understanding UX

We asked in the beginning of the questionnaire how the respondents understand UX in general. UX was understood to be a feeling (28.6%) a user gets when interacting with a product; and the feeling is built on usability (22.9%) (i.e., ease of use and efficiency). Also 22.9% of the 70 respondents emphasized the holistic nature of UX and described it as a whole-life cycle from buying to end of life and said that it includes many aspects. Especially product managers (42.9% of them) and UX specialists (33.3%) referred to the holistic nature, whereas managers (36.8%) and developers (35.0%) described UX as a feeling the user gets while using the product.

Later in the questionnaire we asked what good UX means in the applications or services the participants were developing and got 49 responses. Traditional usability factors were ranked high (Figure 3). Ease of use (34.7%) and efficiency (22.4%) were seen as the most important: 14.3% of respondents thought the service should satisfy user needs, and 14.3% described mainly non-negative user feelings – "*user does not get frustrated.*" Only three respondents (6.2%) described positive feelings, such as "*good UX means that user is happy or likes the application.*" However, more ambitious UX goals are set in the beginning of a project. There is a strong feeling that these goals do not spread in the organization effectively and that people both in the R&D and outside of it are unaware of them. On the other hand, for most of the applications and services the company produces, the product strategy emphasizes invisibility and ease of use.
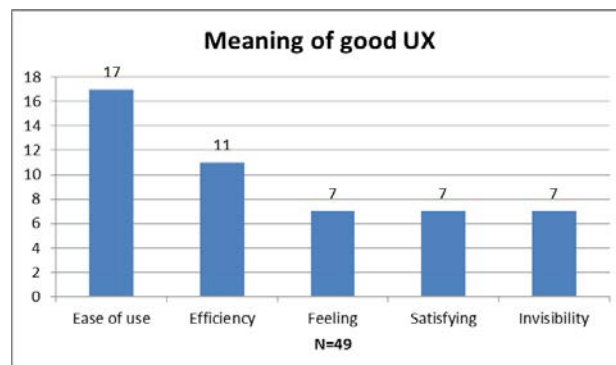

Figure 3. Good UX in our applications (open-ended question)

The company had not defined how they understand UX. Good UX may have various forms depending on context and service. When asked "how does 'company name' define user experience," 37.5% said they do not know; 33.3% described some definition such as ease of use, "*the UX team has defined it,*" or "*no customer complaint means it is ok*"; and 6.3% answered there was no common definition.

5.4 Tasks of the UX Team

The UX team was partly operating as a service organization where product managers or developers could order some design work. It seems that service mode was causing

many of the observed problems related to UX work, such as inability to constantly deliver design in time for development or doing UX work too late. Lund [19] claims service mode is the biggest threat to viable UX work. In the survey, 33.3% of respondents answered that one of UX team's most important tasks is consulting or helping development teams (Figure 4). A UX team member clarified this: *"When a project has remembered they need some specs or screens, then they have ordered those and we made them."* On the contrary, the UX team told in interviews that their biggest responsibilities are making designs and understanding the big picture. Six interviewees wanted to see UX specialists giving guidelines and making high-level designs whereas the developers would solve design details and trivial cases by themselves with help from UX team when needed (H1, H3, H4, H9, H11, H12).
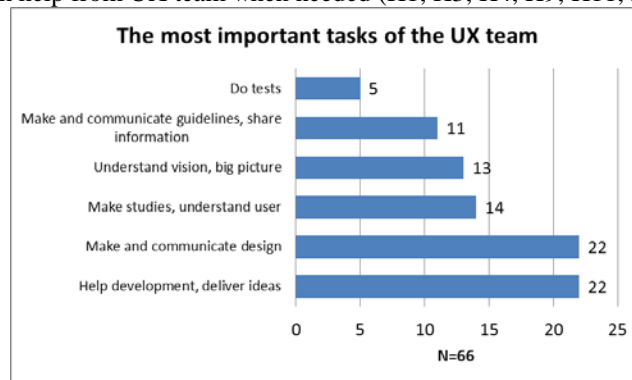

The most important tasks of the UX team

Task	Value
Do tests	5
Make and communicate guidelines, share information	11
Understand vision, big picture	13
Make studies, understand user	14
Make and communicate design	22
Help development, deliver ideas	22

N=66

Figure 4. In the survey, the most important tasks of the UX team were considered to be to consult and design (open-ended question)

The UX team seemed to have limited power of decision: *"One of the most challenging issues in our work is that we have lots of responsibility, we should manage the overall picture of all the requirements and how it can be captured in the UI level. And on the other hand, we have no power of decision at all; we need to be diplomatic with everyone to find the happy medium"* (UX specialist). When we presented the results to company employees (mainly product managers, product owners, UX specialists, and architects), they considered understanding vision, the overall picture, and user should be more important.

5.5 Cooperation with UX Specialists

In general, the UX team operates as a link between R&D and product management. When asked '*what is expected from you in your job?*' an interviewed UX specialist told "*it is mostly communication, acting as a link between development and product managers. And to visualize both business and technical requirements and try to transfer those into pleasant, easy-to-use user interfaces.*" Physically the majority of UX specialists are co-located with product management and apart from R&D. Survey respondents considered in an open-ended question that at its best UX cooperation is started early (18.6%), and communication happens face-to-face (15.3%). However, 11.9% reported cooperation would be at its best as long as it actually happened, or if

they just had a UX specialist available. It seems there are problems with availability of UX resources.

In contrast, unsuccessful cooperation (Figure 5) is conducted too late (26.7%), there is too little or no communication (18.3%), or the design the UX team delivers is unsuitable or not implementable (18.3%). Unclear responsibilities or disagreement on who can make decisions hinders cooperation (13.3%). Too little communication between organizational units may lead to such non-Agile practices as working separately and communicating ready-made work between units (e.g. between UX and R&D). In general, more agile approach for UX work was hoped for (H1, H2, H3, H4, H7, H8, H9, H10, H13).

Participants from R&D generally wished for more face-to-face time with the UX team (Figure 6), as observed both in the survey and interviews. Synchronizing R&D and UX work was seen as a major problem as UX and R&D frequently work on different schedules. Interviewees from R&D mentioned getting contributions from the UX team (e.g., for design) takes too much time. In addition, UX and R&D should cooperate more frequently in an iterative manner. The interviewed architects hoped that architecture and UX design could be done in iterative cooperation.
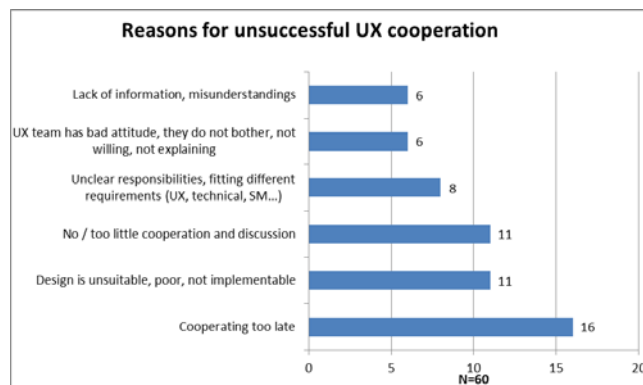

Figure 5. Unsuccessful UX cooperation happens too late (open-ended question)

Often developers are introduced to the design only when implementation starts. Typically, product management outlines the first draft of a concept, and often the UX team is involved, whereas a development team usually joins the work later. "*When the development team gets involved, they just say 'no!' We should have discussed earlier.*" Similar problems occur when product management makes first drafts with developers and without UX team, "*Sometimes all parties are involved (from the beginning), but sometimes product managers work directly with developers and we (UX team) are like 'no,no,no! UX suffers'.*" Interviewees described there is no guidance on cooperation available; individual product managers decide if they want to include UX issues, and UX work can be easily cut out (H1,H3, H5, H6, H10, H12).
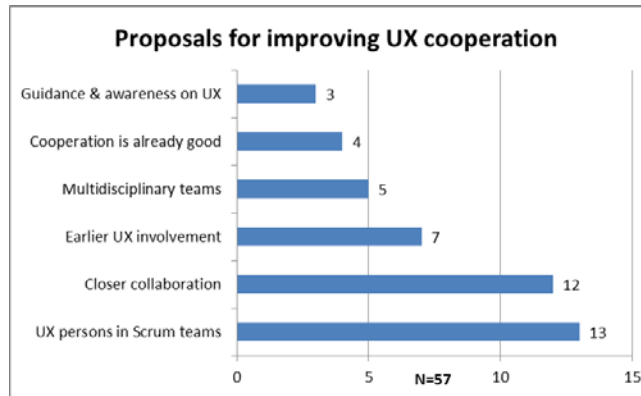
Figure 6. Respondents hoped for more constant and frequent cooperation with UX specialists (open-ended question)

In many cases, architecture designs are made without feedback from the UX team and architects have to "guess" how it should be: "*It's better to have some UX input before final design.*" Interviewees said that the UX team gives feedback on ready-made architecture design and says that this should have been done differently. The same occurs with UX design: "*It kind of feels that we just get some ready-made (UX) design at some point and this is how it looks, and there's no chance to affect that.*" Both interviews and survey indicated that when there are incompatibilities between UX and architectural design, UX and architects discuss changes to architecture with the product owner and product manager. Changes are made if there is time, but usually architecture stays as it is.
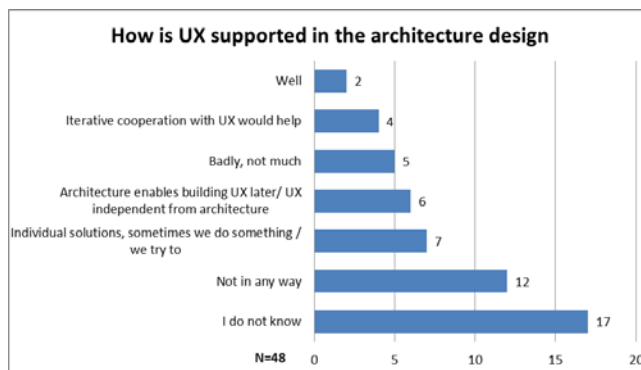

Figure 7. UX is not present in architecture design (open-ended question)

Architecture may be designed without the UX team's contribution even in those cases where contribution would be needed. Over third of survey respondents (35.4%) did not know how UX is supported in architecture design (Figure 7). One fourth (25.0%) of respondents say it is supported minimally or not at all; 14.5% describe their own informal solution; 12.5% believe good architecture enables building good UX later on or that UX is not affected by architecture; and 8.3% hope to have iterative cooperation between UX specialists and architects in the future.

There are no commonly agreed-upon practices in, for example, communicating design or documenting user requirements. Depending on the development team, the UX team may deliver design e.g. as wireframes or highly detailed Photoshop images. Again, at worst, ready-made UX designs may be communicated via email to developers. User requirements may be communicated as user stories or use cases (34,5%), or by various other means. Of the survey respondents, 54.5% told user requirements are documented informally, poorly, or not at all. Interviewed architects and developers hoped user requirements would be incorporated and explained in user stories. Interviewed UX specialists told they would prefer doing wireframe that developers would complete by following the style guide. Doing detailed Photoshop images was seen as waste of time since the details can be seen from the style guide. However, currently it is not ensured that UX will be implemented according to the style guide, or as designed. The interviewed UX specialists hoped that developers or quality assurance would ensure the UX implementation.

6. Discussion

In general, the company had a functioning and competent UX team with established practices. However, the company had divergent ways of working, and UX work was not included in the development process and practicalities sufficiently enough. There were many unofficial practices concerning UX work. Individuals could have made their own decisions whether to follow those practices in their work. Especially decisions made by individual product managers and product owners had a significant effect on the impact of UX work in the company.

Currently, UX specialists had an iterative way of working where they did the design work based on an existing product backlog. They were rarely involved before the PB was created. In some cases UX specialists had some design upfront time. But regularly they also had to start the design work only after implementation work had been started, which they considered stressful. UX design was communicated to developers with various methods depending on developers and the skills of the UX specialist. The UX team also had the role of quality assurance in UX issues as no one else ensured that implementation follows the style guide or the UX design. One explaining issue might be that half of the quality engineers who participated in the survey considered their expertise in UX issues low.

6.1 Identified Problems and Potential Solutions

Table 4 presents problems the organization had with UX issues and suggests solutions to fix those. The solutions are based on related work and the development and business models utilized in the organization. Thus, suggestions from literature are compounded with the present state of the organization. The organization used the suggested solutions to evolve their current practices and to create new better ones. Mainly, the issues presented in the table are due to unclear practices and responsibilities, and the service-organization approach that was prevalent in the

company. The organization already has a functioning and efficient UX team; the majority of UX work already leads to a good outcome. These practices are part of continuous improvement activities and a means to further systematize UX work.

Table 4. Suggestions for observed UX challenges.

Problem	Solution
UX work is started too late	UX specialist involving roadmap concept creation and UX related decisions (early involvement suggested by e.g. [4, 8, 10, 25])
UX design does not match architecture design	Both UX and architecture design made iteratively with both disciplines involved [4, 22]
Developers misunderstand design, UX team does not notice technical limitations	A developer participating in UX design [4, 22]. The participating developer as the first contact person for other developers
UX specialists have to hurry with design	Design made in UX sprints with the help of an architect or developer [4, 22], one sprint ahead of development [4, 10, 21, 27]. Some design upfront [11, 25]
UX design is not ensured during development	Acceptance criteria for UX issues; team owns UX implementation, PO or quality engineer approves [22]
UX work is bypassed because of tight timeline	Criteria for minimum/desired UX at project start [30]
UX elements dropped arbitrarily during implementation	Plan for minimum UX design realization; essential and optional parts or solutions [29]. Power of decision for UX people [6, 11]
UX team doing lots of reactive work	UX team involved earlier [4, 8, 10, 25]. UX specialist involved in decisions when and where UX work is needed [6, 11]. UX team not a service organization [19]

6.2 Implementing Results in the Organization

A workshop reviewing the study results was arranged in the company after the study period during fall 2011. The 14 participants were from different areas of product development, both from R&D and outside of it (3 product owners, 2 architects, a scrum master, a quality engineer, 3 product managers, 3 UX specialists, and a director responsible of R&D methods). The workshop aimed at generating ideas for concrete actions that could be taken to improve the current situation. The participants worked in three groups of four to five persons, and they were instructed to create seven realizable ideas and select three of them to present to others. All the groups wanted to ensure that UX design is realized during development, and therefore suggestions were limited to seven. Finally, the presented ideas were ranked by voting. The ranked actions were as follows:

1. Product Backlog items will be groomed for UX before development starts.
2. Product, architecture and UX roadmaps will be synchronized regularly.
3. UX target will be set in the beginning.
4. Release UX quality will be fast-checked during development.
5. Design version control will be improved.

6. UX resources will be increased.
7. UX stakeholder will be in action during project.

Later, a UX review method was implemented in the official development process. With help of others, the UX team created the method to tackle the most severe issues found in the study presented in this paper: The company started a monthly practice where a user study is arranged to examine a selected feature or product. The participant roles are usually two UX specialists, UX manager, a product manager, a product owner, an architect or developer, and a user. The user has pre-defined tasks to perform, and the others are observing and making notes of certain issues. After the user test part of the meeting, the test is revisited task by task and everyone marks on a whiteboard the biggest flaw they observed. The observations are discussed and ordered. The product owner and architect or developer orders the list by business value and developer effort. The product owner selects how many of the list items they can commit to within the next release cycle. The first experiences of the method show that the method helps to reveal UX flaws and increases the participants' understanding of UX. However, the fixed release plans strongly limit the agility of R&D. In one UX review meeting that a researcher was observing during spring 2012, the product owner said they are able to welcome none of the items to the product backlog since they are already too busy with the release.

7. Summary and Conclusions

In the study described in this paper, we conducted a present state analysis concentrating on UX within Agile software development in a large software organization.

76 employees participated in a large, mainly qualitative survey and 13 persons with different work roles were interviewed. The data were analyzed with a qualitative content analysis method, and with SPSS. Results were divided into categories based on the survey structure, that is, into process, cooperation, and understanding of UX. Another category, the UX team, was generated during analysis mainly to describe the team's current working mode and position in the organization.

The organization is advanced both in Scrum and UX work; they have a competent UX team and knowledge about both Agile and UX development is at high-level in the organization. However, the organization has a separate product management (including the UX team) which operates by yearly and quarter plans, and R&D which follows Scrum. Differences in schedules and ways of working make cooperation more challenging. The most pervasive challenges were related to communication problems, too late UX work, and issues in power relations and in areas of responsibility. Communication issues, caused mostly by too little face-to-face time, led into misunderstandings and time wasting. UX specialists' insufficient power of decision enabled e.g. product owner, product manager or development team to cut out UX issues generally at any point. Responsibility issues seem to be common in Agile UX as the role of UX specialists is unclear in many cases.

The results are not conclusive in the sense that they are to a great extent addressing organizational rather than methodological issues. Consequently further research in this field is needed in order to refine Agile methodologies to better support UX related issues.

8. Acknowledgment

9. References

1. Agile Alliance. Manifesto for Agile Software. Available at http://agilemanifesto.org. (2001)
2. Beyer, H., and Holtzblatt, K. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann. (1998).
3. Boivie, I., Gulliksen, J., and Göransson, B. The lonesome cowboy: A study of the usability designer role in systems development. *IWC 18*, 4, 601-634. (2006).
4. Budwig, M., Jeong, S., and Kelkar, K. When Usability met Agile: A case study. In Proc of the 27th international conference extended abstracts on Human factors in computing systems (CHI EA '09). ACM, 3075-3084. (2009).
5. Bygstad, B., Ghinea, G., Brevik, E. Software development methods and usability: Perspectives from a survey in the software industry in Norway. *Interacting with Computers 20,* 3. 375-385. (2008).
6. Chamberlain, S., Sharp, H., and Maiden, N. Towards a framework for integrating Agile development and user-centred design. LNCS 4044, pp. 143–153. (2006).
7. Chow, T., and Cao, D.B. A survey study of critical success factors in Agile software projects, *Journal of Systems and Software*, *81* I 6, 961-971. (2008).
8. Coplien, J.O., and Bjørnvig, G. *Lean Architecture: For Agile Software Development*. John Wiley and Sons. 376 pages. (2011).
9. Earthy, J. Usability maturity model: Processes. http://www.idemployee.id.tue.nl /g.w.m.rauterberg/lecturenotes/Usability-Maturity-Model[2].pdf (1999)
10. Federoff, M. and Courage, C. Successful user experience in an Agile enterprise environment. *LNCS 5617*, 233-242 (2009).
11. Ferreira, J., Noble, J. and Biddle, R. Up-Front interaction design in Agile development. *Lecture Notes in Computer Science 4536*, 9-16. (2007).
12. Ferreira J., Sharp H. and Robinson H. User experience design and Agile development: Managing cooperation through articulation work. *Software Practice and Experience*; *41,9*, 963–974. (2011)
13. Fox, D., Sillito, J. and Maurer, F. Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. *Proc. AGILE 2008* conference. IEEE Press, 63-72. (2008).
14. Hussain, Z., Slany, W. and Holzinger, A. Current state of Agile user-centered design: A survey. *LNCS 5889*, 416-427. (2009)

15. Innes, J. Why Enterprises Can't Innovate: Helping Companies Learn Design Thinking. In *HCII 2011, LNCS: 6769*, 442–448. (2011)

16. ISO 9241-210:2010. Ergonomics of human-system interaction. Part 210: Human-centered design for interactive systems (2010)

17. Kuusinen, K., Väänänen-Vainio-Mattila, K. How to Make Agile UX Work More Efficient: Management and Sales Perspectives. Accepted to NordiCHI 2012

18. Lazar, J., Feng, H.F., and Hochheiser, H. *Research Methods in Human-Computer Interaction*. John Wiley and Sons. (2010)

19. Lund, A.M. Creating a user-centered development culture. *interactions 17*,3, 34-38. (2010).

20. Marcus, A., Ashley, J., Knapheide, C., Lund, A., Rosenberg, D. and Vredenburg, K. A survey of user-experience development at enterprise software companies. *LNCS 5619*, 601-610. (2009).

21. Miller; L. Case Study of Customer Input for a Successful Product. Proc. of Agile 2005. Agile Alliance. (2005).

22. Petrovic, K. and Siegmann, M. Make space for the customer: The shift towards customer centricity. *LNCS 6769*, 485-490. (2011)

23. Runeson, P., Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering, 14*, 2 2009.

24. Scrum Alliance http://scrumalliance.org/

25. Silva da Silva, T., Martin, A., Maurer, F. and Silveira, M. User-centered design and Agile methods: a systematic review. *Proc. of the International Conference on Agile Methods in Software Development (Agile 2011)*.

26. Sutherland, J. and Schwaber, K. The scrum papers: Nut, bolts, and origins of an Agile framework. Available at: http://jeffsutherland.com/ScrumPapers.pdf (2011).

27. Sy; D. Adapting usability investigations for Agile user-centered design. *Journal of Usability Studies 2,* 3, 112-132. (2007).

28. Venturi, G. and Troost, J. Survey on User Centred Design integration in the industry. In ACM International Conference Proceeding Series; Vol. 82. *Proc. of the third Nordic conference on Human-computer interaction*. (2004)

29. Venturi, G., Troost, J. and Jokela, T. People, organizations, and processes: An inquiry into the adoption of user-centered design in industry. In *International Journal of Human Computer Interaction 21*, 2. (2006).

30. Viikki, K. and Palviainen, J. Integrating Human-Centered Design into Software Development - An Action Research Study in Automation Industry. *Proc. of* (SEAA'2011), IEEE Computer Society. (2011).

31. Zhou, R., Huang, S., Qin, X. and Huang, J. A survey of user-centered design practice in China. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 1885–1889. (2008).