

User Experience Design Goes Agile in Lean Transformation – A Case Study

Minna Isomursu, Andrey Sirotkin

VTT Technical Research Centre of Finland
Oulu, Finland

minna.isomursu@vtt.fi, Andrey.sirotkin@vtt.fi

Petri Voltti, Markku Halonen

Nokia Corporation
Oulu, Finland

petri.voltti@nokia.com, markku.halonen@nokia.com

Abstract— This paper describes the results of a single-case case study, exploring the role of user experience (UX) work in agile software development. The case study company is a large multinational telecommunication company undergoing a lean transformation process. In this case, lean transformation includes the adoption of agile software development practices. Transformation to agile practices had taken place one year prior to the analysis. The analysis is based on documentation analysis and semi-structured interviews of seven software development professionals. The results show that there were difficulties integrating UX design and software engineering work in an agile and iterative manner. The transition process succeeded in shifting UX and related documentation to a central planning role. The roles of the UX designers in the teams were still under re-definition. There was also a clear need to establish new ways of collaboration between UX professionals and software designers.

Keywords—lean transformation; agile; user experience design

I. INTRODUCTION

This paper summarizes and analyses problems and practices of user experience (UX) design and related development activities in a company undergoing a lean transformation process. The lean transformation process aims to reorganize software development work in a large multinational telecommunication company. The development process used in the company was previously based on an incremental development process. As a result of lean transformation, it was changed to follow the principles of agile development. The new agile way of working has shifted UX to a central concept for planning, designing, monitoring, implementing, verifying, and validating development activities and results. Issues related to UX have therefore become central to everyday development work.

The goal of the paper is to discuss and analyze difficulties that can arise in adopting agile working practices looking specifically on integrating the work of UX designers with other software design practices during the transformation process. The analysis is based on a case study where the transformation process of one particular company is explored. The paper is structured according to the case-study guidelines by Runeson and Höst [32]. It first summarizes previous research into integrating software development tasks related to UX design, such as user-centered design (UCD) and user interface (UI) design, with agile software development and the problems that have been

reported to emerge in this context. The research process and methods used in the case study presented in the paper are then described in detail. The paper then progresses by presenting the case study findings. Finally, discussions of the findings and conclusions are presented.

II. RELATED WORK

The problems and possibilities of integrating UX design into agile work practices have been studied by many research groups during the last decade. Most studies have explored integrating usability [7, 9, 33], UI [14], or interaction design [14], with agile software development, or how UCD, in general, can be used in agile software development [4, 6, 11].

During the last decade, human-computer interaction (HCI) researchers have directed growing interest towards “user experience” research to emphasize the shift from a task-oriented, cognitive concept of usability towards the direction of a more holistic, emotional, and subjective exploration of the responses technology has from its users [16]. Most of the UX research discusses what UX actually is and aims to define UX as a concept or recognize components that are relevant to UX [24]. Some research work has also been directed at exploring the methods for UX evaluation and design (e.g. [4]). Very little research has been conducted on professionals in UX and the way tasks related to UX are performed and organized in SW development organizations.

Here, UX design is defined to comprise all activities performed in software development to ensure the desired target level of UX in the final product. The most visible part of the product to the user is the UI, so, UI design, together with interaction design, plays a central role in UX design. Focusing on UX, however, also makes UX targets central planning tools for other software engineering professionals. For example, managers need to ensure UX targets will be met, test engineers need to use UX targets in planning etc.

Usability engineering ensures that the end product is easy to learn and use, and it therefore addresses one important contributor for the whole UX. Usability testing has often been named as the most effective UCD method [9]. UCD is used as a holistic umbrella term for a collection of practices that aim to produce systems that meet the needs of real users and are highly usable [4]. UIs, interaction design, and usability engineering can therefore be seen as methods that can be used in UCD. As user experience is a holistic product experience [27], UX work needs to be holistically integrated into the work of the whole development team. UCD is one

commonly used method for ensuring that the user focus is at the center of the development activities [9]. UI and interaction design, and agile development have been analyzed and found to have much in common. Both emphasize the importance of iterative development [6, 18] and stress the role of collaboration with different stakeholders, including customer and business professionals, in the development work [4, 14]. Both support rapid feedback cycles and testing [14], and accept, and even embrace, change as a natural phenomenon [4].

A. UX design in corporation

Lean and agile methods contribute to a company's ability to conceive and implement a value-creating strategy. The lean philosophy focuses on increasing value by removing waste [39], while agile principles emphasize the need to search for and take advantage of market opportunities in a volatile business environment [36]. The implementation of the two concepts in a company is motivated by the need to deliver added value to customers, which is also one of the basic principles of a corporate strategy. The misalignment of strategic and operational goals and activities may therefore impede product development and threaten competitiveness.

UX at company level is a strategy for product differentiation. From the company's perspective, UX is a way to sustain a competitive advantage by delivering value through unique memories and experiences [17]. On the user side, UX is the way users perceive and attribute value to a product. Thus, UX is a binding element of the company's and the user's perspectives [15]. The interconnection between the targeted UX and operations management with the development objectives is necessary for a company's success [12]. This interconnectedness creates a capability for a sustainable competitive advantage [36].

Capabilities channel organizational resources in the process of value creation [2, 30]. They are enablers for creating and maintaining a competitive edge by taking advantage of market opportunities [3, 36]. In a volatile telecommunication market, investment in capabilities will not only sustain the competitiveness of the company but also develop its ability to shape the market [8]. One of the reasons for telecommunication companies to master their market-shaping abilities is the effect of cloud technologies. Despite many challenges, it transforms software and telecommunication industries [5, 12] by shifting the emphasis from reliance on the ability to identify existing market opportunities to a need to shape market demand. Furthermore, a company's innovation, value capture, and financial performance depend on its ability to shape resources in its ecosystem [31]. As a result, the company's survival is largely determined by its capacity to deliver valuable experiences while excelling in orchestrating access to multiple resources. This case study reviews UX design in the software development process. The management can attain further insights into the UX design capability by viewing the findings in the wider context of the company's ecosystem. This is a necessary exercise, because the UX design requires coherent integration of processes throughout the levels of the organization [12]. This means that the UX

design value for the end-users hinges on the company's ability to interweave its "sensing" of market opportunities all the way to the development process.

B. Integrating UX design with agile practices

Several challenges have been identified in previous research on integrating UX design and related activities into agile software development. One of the most commonly reported problems deals with finding a balance between up-front interaction design and integrating interaction design with iterative coding with the aim of delivering working software instead of early design concepts [6, 14]. Dayton and Barnum [9] report that usability testing is easily overridden by urgent development goals of sprint cycles. This has also been observed by Blomkvist [4] who states that usability can easily fall by the wayside, as agile development is primarily aimed at efficient coding tactics together with project management and team organization instead of usability engineering. He continues with the observation that a lack of professional usability is a weakness in most agile processes. The same observation was made by Jokela and Abrahamson [20] in an investigation of agile software development. Traditional usability methods are often judged too heavy and slow to be integrated effectively into agile ways of working. Need for more simple and low-cost methods is recognized [21]. Examples of proposed methods include Rapid Contextual Design [3] and Discount Usability Engineering [21]. Also, the durations of UI design and coding iteration cycles have been found to be different: UI design cycles are very rapid, lasting for hours or days, whereas agile coding iteration cycles typically take a few weeks [14]. This can make scheduling and coordination of work difficult. Even though agile methods emphasize the role of the customer, it is rare for real end-users to take on the role of the customer [6]. UCD practices can therefore be overshadowed by concentrating on the customer instead of the end user.

Even though there are many challenges in integrating UX design with agile practices, some researchers see agile software development practices as enablers for bringing UX design closer to software engineering and enhancing interaction between these two disciplines [28]. A survey of the integration of usability and UCD practices with agile methods published in 2009 [19] reported that the majority of the respondents found that usability and user-centered design practices had brought added value through improvements in the usability and quality of the end product. Development teams often report that they are better able to respond to the needs of the customer with agile methods [13], and their measured or perceived productivity has been reported to be better than development teams using traditional methods [13]. Some studies indicate that traditional software engineering teams are perceived to develop better and more consistent UIs than teams working with agile methods [38]. Previous research conducted in the same organization as that studied in the research reported in this paper shows that people, in general, are satisfied with agile ways of working and would not want to return to their old development practices [23]. The adoption of agile practices has also been greeted with satisfaction in other organizations [13, 26].

III. CASE STUDY DESIGN

The case study presented in this paper was conducted during the first half of 2010 at Nokia, a large multinational company whose telecommunication products embedded a large number of software components. The case study is descriptive and interpretive by nature, i.e. it aims at portraying a situation through participants' interpretation of their context [32]. The research examines what challenges and critical issues may arise in a software development organization in UX related practices when the organization undergoes a transition from traditional iterative software development into agile working practices. The research process was divided roughly into four phases: (1) case selection, (2) planning, (3) data collection, and (4) analysis. The next sections explain the methods and procedures used in each research phase. Finally, the terminology used is described, and the limitations of the research are discussed.

A. Case selection

The case organization Nokia represents a company whose software intensive products are developed in globally distributed team for global consumer markets. The development project explored is a typical feature phone development project, i.e. the end product is a consumer product targeted for wide range of global users. The organization has practiced systematic software development assessment and improvement activities from early 90's, and they have resulted in several incremental and transformative systematic changes in product development practices. The case study explored a development project that had adopted agile development practices one year before the analysis. The adoption of agile software development practices was one step towards the lean transformation that was ongoing in the organization. The development teams were distributed across three continents in Northern America, Asia, and Europe. The analysis presented is based on the interviews of the seven research subjects who represented roles essential to the UX design and implementation. All were experienced software professionals who had experience both on old working practices and the new agile practices. The interviewees included: (1) software project manager with eight years of experience of software development projects, (2) UX leader with over 4 years of experience of UX work, (3) test manager with four years of experience of the product

development process, (4) UX designer with three years of experience of UI design, and (5) three members of a SCRUM team responsible for implementation activities.

B. Planning

During the planning phase, the people involved in data collection were trained in the assessment procedure used. The assessment procedure was tailored to meet the requirements of the organization, and it was based on Capability Maturity Model Integration (CMMI) by the Software Engineering Institute (SEI), Carnegie Mellon University. Training included defining the ethical code for the research. Confidentiality issues were planned with care; it was important to make sure that interviewees could be positive that their confidentiality was not risked either internally in the organization or externally in reporting the results. Failing in conveying the atmosphere of full confidentiality would risk the reliability of the results. The data collection was planned to include documentation analysis and semi-structured interviews. The selection of representative projects and teams was made and an interview schedule drafted. The interview team consisted of the lead assessor from the company accompanied by a group of four assessors also from the company. Three researchers also took part in the data collection process. The interviewees were selected by the lead assessor through discussions with people involved in the development project.

C. Data collection

The data were collected through: (1) documentation analysis and (2) semi-structured interviews. The documentation analysis was performed by browsing the electronic collaboration tool environments used by the development projects. The goal was to become familiar with the terminology and to see if common templates or methods were used in the UX documentation. As all the product-related documentation was confidential, the documentation analysis was restricted to browsing the documentation at the company premises. After the documentation analysis, semi-structured interviews were conducted. The conceptual framework used in the interviews is described in Table 1.

The interviews were organized in the context of a wider process assessment, focusing on the assessment of the status and problems related to lean transformation that was ongoing in the organization.

TABLE I. INTERVIEW FRAMEWORK

Focus	Description
Role in UX design	How is UX design visible in your work? What is your role in UX design?
Interfaces	Where do you obtain input for your activities related to UX design? How and to whom do you report or communicate your work related to UX design?
Documentation and communication	With what kind of documentation dealing with UX issues do you work? Design and evaluation How is UX described or modeled in this documentation? What kind of communication is used during the development work on issues related to UX?
UX mindset	Do you think that UX targets and descriptions are clearly visible in all the development work?

The results presented here are based on a subset of information collected during the assessment process. This means, that the findings presented in this paper are derived from the interviews of software professionals whose work was closely related with UX activities (described in chapter 3.1), and analysis of UX documentation. However, the lead assessor had taken part in all interviews and data collection activities, he could contribute to the analysis with the findings from the larger body of interviews.

For practical reasons, the duration of each interview was constrained to approximately one hour. All the interviews lasted for at least one hour. With the software project manager, an additional interview time slot of one hour was reserved, as not all the things could be covered during the first interview. As the interviewees were located on three continents, video and telephone interviews were used.

D. Data analysis procedures

The data have been analyzed through the process of qualitative content analysis [22]. The UX evaluation part of the assessment was the responsibility of one researcher who took part in the interviews of all seven subjects (as described in chapter 3.1). The researcher drew up summaries and extended scripts of the interviews, and shared her notes with the others assessors. Two other members of the assessment team, the lead assessor and the coordinator of the lean transformation process, took part in the analysis process by validating and complementing the results through their observations. The interpretation phase of the analysis process therefore links the analysis to the greater scope of the lean transformation assessment, as the lead assessor took part in all 45 interview sessions involving 57 individuals throughout the assessment process.

E. Terminology

The product development in the company is organized as a software factory in which the development work is organized to deliver components and features that can be integrated into products in product programs. Here, our focus is on the projects that develop components and features and deliver them to product programs that are customers of the assessed projects. The software development is organized based on the principles described in a white paper by Leffingwell & Aalto [25], and the terminology used in this paper follows these high-level principles.

The highest portfolio-level requirements are called epics. Each epic is specified in more detail as an experience. Experiences are further specified as ambition levels of epics that specify work that can be delivered within one experience development train (EDT) and user experience targets that specify a target for the work done in the specific EDT. Product-specific UX targets are also defined in product programs. The work to be done in sprints within an EDT is planned in user stories that should be based on UX targets and ambition levels. User stories finally generate tasks for scrum teams. Timebox programs take care of releasing epics to specific products through releases of EDTs. The work within the EDT is organized in sprints that are executed by scrum teams. The terminology and most important planning

documents are illustrated in Figure 1. The boxes in the figure represent process entities (deliverables or actors) and the arrows represent “defined by” relationship, i.e. one entity (target of the arrow) being defined by or based on another entity (source of the arrow).

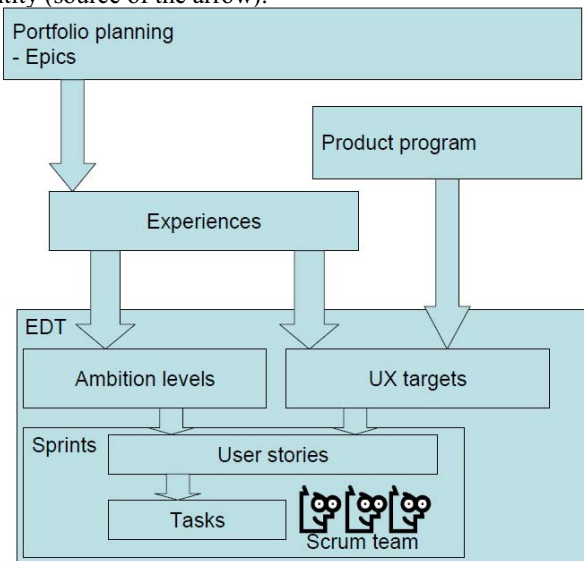


Figure 1. Terminology and most essential UX documentation

IV. LIMITATIONS AND VALIDITY OF THE RESULTS

The research method chosen in this research, i.e., single-case case study, raises a question on the generalizability of the result. The primary goal of the case study presented here is to provide a detailed description of a relatively novel research target, i.e., how to organize and conduct UX design in agile software development. The case study provides an opportunity to learn about an environment that is difficult to access, i.e., the confidential software development activities of a multinational company. External validity is primarily achieved through a detailed description of the target environment and by grounding the previous research in this area.

Internal validity has been tackled with three types (Denzin’s typology for triangulation [10]) of triangulation. First, data triangulation was used in the data collection, as various data sources in terms of people working in different roles and at different sites were interviewed. Second, methodological triangulation was used by applying two (document analysis and semi-structured interviews) complementary data collection methods to obtain evidence from different, complementary sources. Third, investigator triangulation was used, as several assessors with different backgrounds participated in the data collection and analysis. They included software development and assessment professionals from the company, and researchers outside of the company.

V. RESULTS

Here, the results are organized according to coding scheme into a structure that arises from the content analysis.

Even though the basic structure follows the conceptual framework used in data collection, some restructuring has been performed, as some topics were emphasized more than others in the collected data and their interpretation.

A. Organization of UX work

The interviewees had a relatively uniform understanding of what “UX work” means. UX work was primarily described through the work of the UI design. UX professionals were seen as the design professionals responsible for the UI design that would then be implemented by the software engineers and tested by the test engineers.

1) Organization of teams

During the course of the development process and the related lean transformation, two different ways of organizing the work of the UX designers inside and outside the scrum teams were tried out. In the first setup, the UX designers formed their own scrum team led by UX leaders who assigned them tasks identified in the implementation plans of the scrum teams. In the second setup, each implementation scrum team had a UX designer as a member of the team. Problems were identified in both settings.

When the UX designer worked as a member of a scrum team, problems were identified of achieving a coherent UX across the development teams and balancing the workload. When UX designers worked as members of the implementation teams, there was a risk that the solutions related to the UX design would start to divert, leading towards a non-coherent product UX. The workload for UX professionals was very uneven, as the effort needed for UX work was very different in the different development phases and between the implementation scrum teams. Both problems were primarily UX management problems: it became more difficult to manage and steer the product UX, coordinate communication between the UX designers, and balance the workload of the UX professionals. On the other hand, when UX specialists were members of scrum teams, they were better able to use incremental and agile ways of working also for UX design, whereas in centralized UX design the same level of agility was not achieved. Both UX designers and software engineers working in scrum teams were happy with the level of integration that could be achieved through this setup.

Another option for organizing UX professionals (here UI designers, graphic designers, and localization professionals) is to have them all in one scrum team responsible for UX-related development work. This provides the UX manager with better control and visibility of the total UX, and better tools for efficient resource allocation. The challenge is to maintain a high level of collaboration and communication between the UX designers and the rest of the implementation team. This was found to be a major challenge, even when the UX designers were allocated directly to the implementation scrum teams. This was tackled by UX designers participating in the sprint planning meetings to identify UX-driven development tasks and then share them in separate UX designer meeting for prioritization and work allocation.

The role of the UX leader was central and important in both ways of organizing the UX work. The role of the UX leader was unclear throughout the development organization. Even though there seemed to be no confusion on what UX work is in practice, the responsibilities on who should take UX target setting leadership, or who should take responsibility on validating and accepting UX results were observed. The responsibilities of the UX leader were not always well understood, and the expectations for them were therefore not realistic. In addition to the organization of the scrum teams, an additional challenge was set by the geographical separation between the UX designer and the rest of the scrum team. The weekly planning sessions provided an opportunity for status checking but not for discussions related to design and implementation. The UX designers at geographically distant locations therefore had little visibility of the actual implementation tasks.

2) Scheduling design and implementation into sprints

UX work is highly design oriented, i.e., it requires design visualizations, demonstrations, mock-ups, etc. before it matures enough for the actual writing of the software code. The UX designer therefore easily becomes “out-of-synch” with the rest of the implementation team. The UX designer can work on designing user stories and related UI proposals a couple of sprints “ahead” of the rest of the implementation team. This way, the team can ensure that the design is ready and validated before it is handed out for implementation.

In practice, this means that the integration of the work on UX design and software engineering is not optimal however. The implementation team works on user stories that the UX designer has finished some time before, while the UX designer works on other issues. The optimum level of interactivity and agility is then difficult to achieve.

3) Collaboration with the development team

The work of many of the UX experts was not optimally integrated into the work of the implementation scrum teams, because of the geographical distance or organization of the UX professionals into separate teams. Many collaboration practices were observed. The UX designers participated in the EDT and sprint planning, and this provided them with good opportunities for defining the UX design tasks. The planning meetings were seen to play a big role in the interaction between the UX experts and the software engineers. If the UX experts were members of the scrum team, they took part in the weekly scrum meetings for status checking. The collaboration tool was used for documentation sharing, and the related discussions were well integrated into the work processes and supported interaction and discussion. The discussion was primarily had between people in similar roles, however, i.e., between UX designers. The tool support for collaboration had clearly not succeeded in solving the communication problems arising from inadequate opportunities for face-to-face communication between UX professionals and software engineers.

4) Separating UX design from implementation

The question of optimal organization of UX design resources and interaction between UX designers and software engineers had also been approached by discussing whether these roles should be separated in the first place.

Would it be possible to integrate UX design tasks into the tasks of the software engineers? This would provide better visibility for the way UX design manifested itself in the actual implementation, and it would help maintain the focus on UX throughout the development. Doubts were also raised with regard to whether the recognized bigger role and focus on UX introduced by lean transformation and agile thinking were actually supported by increasing the resources of the work dedicated to UX, thereby creating pressure to add UX tasks to the work of all software professionals.

B. UX documentation

The UX documentation, and the other software development documentation, was shared with the development team through a common collaboration tool that supported document sharing, version control, and discussion related to documents. The tool was considered essential to supporting the collaboration of the distributed teams, and people were generally quite happy with that. There were ongoing efforts to create and adopt new tools to develop UI design documentation. The new tools would improve the support for incremental development, and they could be used by the whole development team, not just the UX professionals. The company had decided to develop its own UI design documentation tool, as it had evaluated the tools available on the market as not being lightweight enough to support very rapid iterations and collaboration of UI designers and software engineers.

The following sections describe the most important UX documentation when starting the project, during the development work, and at the validation and testing phases.

1) Starting the project

The UX designers receive documentation that can be used in scoping and focusing the work from portfolio planning and other activities carried out prior to starting the actual software development. The most important documentation was considered to be:

- Product briefs that describe the target markets, competitor analysis, etc.
- Description of UX targets and ambition levels

This documentation was generally valued and perceived as useful. The UX target description and modeling, in particular, were generally and unanimously perceived as too generic and qualitative however. Methods for a more accurate and measurable UX target definition would be appreciated. The documentation, especially that completed in the early phases of the project, was often not updated iteratively to meet the evolving nature of the UX targets.

It was generally agreed, however, that the starting point for the development work took the UX into account well with a set target and ambition levels. These were clearly communicated and well understood by everyone who participated in the development work. As the work progressed, however, this message was not reinforced enough. When new people joined and old project members forgot about their initial targets and ambition levels, the clear vision for UX that was present in the beginning did not remain as clear as the work progressed.

UX targets and ambition levels were better used and internalized in product programs than in epic program planning. Product programs used UX targets and ambition levels in planning, while procedures for embedding UX targets and ambition levels into epic planning were not yet optimal or well established.

2) During the development work

UX professionals were responsible for breaking the epic description into experience targets and user stories. In the development process analyzed here, there were more than one hundred user stories, and this is expected to continue to grow, as some user stories still required splitting as the implementation progresses. User stories were then further divided into tasks that can be allocated into sprints. The issues related to UX design for specific tasks were documented in a UI proposal that is usually presented in PowerPoint slide format. There was no formal methodology for constructing UI proposals, though designers have established common practices over time. Even though the document was commonly called “UI proposal”, the examples available through the documentation analysis and given in the interviews in the documentation also included components that covered a wider scope of UX than just the UI of the system, e.g., issues related to the goals of action, feelings or expectations of the user, stories describing use cases and usage situations, etc. The UI proposals were shared between the members of the development team through a collaboration tool that allowed document sharing and provided an opportunity to share comments and discuss issues related to the document. The tool was adopted for documentation sharing. The commenting and discussion seemed to support mostly internal collaboration by UX professionals, however, not collaboration between UX professionals and software engineers. The UX documentation process was planned so that the UX design is not frozen in practice until the software implementation is ready. The synchronization of the software implementation and UX planning is therefore not very agile. In addition to poor synchronization of the UX design and software implementation, insufficient opportunities or procedures for frequent feedback and discussions to negotiate design and implementation decisions were in place.

One problem identified in the development documentation was the lack of design rationale. The decisions related to implementation and design were not always made to achieve the UX targets but for other reasons, such as schedule pressures or a lack of human resources. In these cases, the rationale behind the design decision was not documented, which could cause problems afterwards because non-optimal design decisions were not reconsidered. First, there is a risk that these cases are forgotten and not identified again, as they are not marked for reconsideration. Second, as the rationale is missing, it is extremely difficult to open cases for reconsideration, as facts about the decision and related implementation or design solution are missing.

3) Testing documentation

The testing of the results of the UX design was performed by UX professionals. It included prototype reviews in the early phases and usability tests in the later

phases. A common problem, however, was that the UX testing was not systematic and that there was a lack of clear agreed procedures especially for test planning, testing procedures and responsibilities related to acceptance of testing results. More details on the actual testing practices will be described in Section 4.3. The test cases were derived from user stories by the scrum team. The goal was for the test cases to cover the main points of the user stories. No systematic measures or methods for coverage test cases relevant for UX design were used.

C. Validation and testing

UX validation was clearly divided into two separate phases that had little interaction. First, the UX design was validated by the UX designers. The most important methods in this phase were expert reviews and user tests with simulation visualizations. Second, the actual implementation of the UX design was tested through test cases derived from user stories. The UX designer had little visibility of this second phase of UX testing.

1) Review of the UI proposal

The UI proposal was validated internally by sharing it for review within the development team. A collaboration tool was used to support the validation activities. The UI proposal was shared through a collaboration tool with all the development team members for commenting. In practice, most of the comments came from other UX professionals. More opportunities for expert evaluations in the early phases of UX design would have been appreciated. Currently, early validation activities are limited in scope, and the validation activities are concentrated towards the end of the design and implementation phase. An even bigger problem identified is the lack of interaction inside the development team. The involvement by the implementation and testing experts in the UX design phase for validation was very limited.

2) Testing the UX design

The UX designers tested their UI proposal through Flash prototypes or other simulations that could be handed out for user testing. Demonstration sessions would provide an excellent communication channel with customers from product programs, if representatives of different project levels would participate in demonstration sessions. However, there seemed to be lots of uncertainty and confusion on who should participate the demonstration sessions, and responsibilities and best practices were not yet clear. Participation, especially by product programs, would be needed, as they have the best understanding of the UX targets and ambition levels and should play a leading role.

Core use cases were tested in the target market environment with users from the target segment. Common evaluation frameworks were used in the user testing, which enabled comparisons across user stories within one experience. The evaluation framework consisted of traditional usability testing complemented by the evaluation of subjective experiences. There is a clear need for methods and tools for agile UX testing that can be used throughout the software development cycle. The UX testing should be planned and integrated more systematically into iterative development cycles. Our further analyses show that very

little systematic methodological or tool support for UX testing is available even though they would be highly relevant and needed in practice.

After testing, the UI design was forwarded for implementation. From this point, the UX designer had very little visibility of the implementation or testing activities.

3) Testing the implementation of the UX design

The implementation of the UX design was naturally tested through testing of the integrated product or as part of the product. Test cases were derived from user stories. No systematic indicators for test coverage from the UX point of view or setting of separate UX acceptance criteria were used. Therefore, the specificity of UX design was not really a specific concern at this point. The test cases were estimated to cover the basic functionality of each use case. The test cases were reviewed together by the scrum teams and test engineers. The maturity of the software to be tested was also reviewed before moving into the testing phase.

The testing of the implemented user stories was also performed in the target market with representatives of the target user segment. At the time of the assessment, however, the development process had not yet advanced into the phase where user testing would have been performed. There were some concerns of how well the findings and results of the testing could be fed back to the implementation because of the tight time schedules.

D. UX mindset in steering the project

In principle, one of the main goals of the lean transformation process in this company was to make the UX the central goal of the development activities. This chapter summarizes the practices and issues identified during the study as drivers or challenges in the UX mindset.

1) User stories in a central role

The user stories played a very central role in the design and planning activities. The user stories were clearly used in all the development phases and tasks, and they were constantly updated to represent the understanding of the design problem at a given time. As can be seen in Figure 1, user stories were defined based on UX targets. This was first done at the early phases of development, and UX targets were intended to be used as basis of user stories throughout the development process. However, it was observed that the visibility of UX targets in the daily development work diminished gradually as the development process advanced. Therefore, a question of how well the user stories actually reflect UX targets may arise. There is a danger that as user stories evolve and are subject of agile iterations during the development phase, they start to live their own lives and the connection to original UX targets is lost. Also, as UX targets are, by nature, more fixed and stable compared with user stories, they are not as visible in the daily development work. Therefore, even though pressure to update UX targets would arise during the process, the changes are actually done only at the level of user stories, and the higher level UX targets are not updated to reflect the new understanding.

2) Perfectionism problems

In the beginning of the project, the UX targets were clearly set and written, and the people involved understood

them well. As the project advanced and more new people joined, the UX targets were forgotten and their role in guiding the UX design decreased. A specific problem of the UX targets that were originally set at basic or low ambition levels was observed. The targets originally set at basic or low ambition levels started to deviate from the original ambition level. New targets were typically set higher than the original targets. It seemed especially difficult to adhere to the targets set at the lower or basic levels, as the developers seemed naturally inclined to strive for the best possible UX. As the scheduling and resource allocation were originally done with lower ambition levels, changing the ambition levels to achieve higher UX goals often resulted in challenges in timetables and resourcing. As a result, even the ambition levels that were originally set as low or basic were difficult to achieve or, even worse, the UX targets with high ambition levels had to be compromised.

3) *UX-roles unclear*

People, especially those carrying out UX management or design, felt that different roles and their responsibilities in UX design and related activities were unclear in the organization. They felt that the expectations of the UX experts did not always meet their actual responsibilities, and sometimes their actual responsibilities did not give them the opportunities to steer the implementation so that the UX actually played the role it should in development-related decisions. The lack of visibility of implementation activities taking place after UX design had been forwarded to the implementation team was especially problematic. As the plan evolved in agile fashion, the UX designers felt that they were not able to contribute to changing plans even though they felt that they had the holistic responsibility of the UX related decisions also during final implementation phases.

Many of the UX-related roles were either completely new or old with changed responsibilities. Their integration into the software development and testing work processes was still ongoing. Our observations indicate that introducing new roles and responsibilities in the organization takes time. In this case, people working in UX related roles felt that their roles and responsibilities had to be re-examined iteratively to ensure that their responsibilities matched the expectations the development team had. This was a process that unfolded only through action; the roles and responsibilities could not be fixed at early stages and then left static.

4) *UX leadership*

The product development organization puts product programs into a central role in processing and internalizing UX targets and ambition levels into the final products. Their role in UX leadership should therefore be strong, as they should represent the “voice of the customer”. However, it was observed that product programs did not always have enough tools and procedures to influence the planning processes to enforce UX leadership. There were not enough opportunities for product programs to see or comment relevant plans in early phases and the product program key persons seemed to be not very well informed about design related decisions taken in early planning phases. This leads to a situation in which the UX leadership is not clear, or is not steered clearly enough by product programs that are

customers for agile development and should be the voices of the customer throughout development phase.

E. *Summary of results*

The analysis shows that, generally speaking, the adoption of agile working practices had placed user experience design into a central and important role in development work and documentation in this organization. User stories were central tools in design and work planning. They were used in daily work planning and communication by all developers regardless of their roles and background. However, the UX targets that should define user stories were not always visible enough in daily work to clearly drive UX related decisions. Their role and visibility were considered to be at a very good level at the beginning of the project, but their role and visibility decreased over time. Communication and the actual active reflection of UX targets in relation to the development decisions made throughout the development process reinforce the visibility and value of the defined targets. The UX targets were clearly communicated, and they were well understood at the start of the development work, but they became less visible and their role decreased as the development work progressed. As the uncertainty relating to the details of the development activities unfolds, as the development work progresses, the UX targets need to be refined to be more accurate, detailed, and explicit to reflect better the achieved understanding of the design problems to achieve an optimal steering effect. The ongoing lean transformation process had not yet fully reshaped the development roles. The new increased focus on UX was clearly visible through the working practices, though not to the same extent as the roles and responsibilities of the people. The role of UX design had not stabilized, and much uncertainty of how to organize the UX design was observed. This manifested itself in communication problems and difficulties integrating UX work with other design and implementation activities. Re-definition of UX design related roles according to the roles and responsibilities used to form and manage scrum teams seemed to be not yet complete, resulting in confusion on responsibilities of UX designers. It was not clear, if UX designers should set, monitor and accept results as the product owners in scrum teams do, or should their role be more of a facilitator who communicates UX targets and works on gradually building UX design towards the common UX goals.

VI. DISCUSSION

Our findings indicate that the adoption of agile software development practices will trigger a need to reconsider the role of UX professionals. This is well aligned with findings of previously published case studies [35]. In addition to the need to redefine the roles of UX professionals in development teams, our results show that communication by UX professionals with the development team is not considered optimal by either the UX professionals or the software engineers. Our findings emphasize the role of the organization of UX professionals, as members of agile teams, and the difficulties caused by geographically distributed teams, as causes of communication problems,

whereas other researchers have found evidence of power aspects between UCD professionals and software engineers that hinder communication [6]. Better team interaction and less time invested in elaborate design, both characteristics of agile development, have been suggested to reduce the need for defensiveness and territoriality of design [1].

The projects analyzed in this research had tried to have a centralized UX team and distributed UX professionals as members of software development scrum teams. Both solutions had advantages and disadvantages, and it was not clear at the time of the analysis which method would be adopted as the common practice, should this be decided on case-by-case basis, or would a combination of both models be an optimal solution. Our results suggest that centralized UX team is preferred by UX managers as it makes management easier and give UX manager more control on the work of UX professionals. On the other hand, UX professionals and software engineer may value decentralized UX work as it give better visibility and opportunities to influence the design decisions early during the development process. Therefore, selection of which option to choose depends on how these viewpoints are balanced in a certain setting. Centralized model allows more control in situations where a holistic UX is a priority, and distributed solution allows more agility in situations where it is needed. Also, even in distributed model centralized UX leader can be appointed to check and monitor holistic user experience at the product level. Previous research shows that the UCD staff is usually centralized, and there is an indication that centralized organization also emerged as a predictor of perceived UCD effectiveness [41]. Our results propose that, with shifting UX as a central planning tool and goal, UX design needs to be better integrated into the work of all software professionals, however, and it cannot be left only to dedicated professionals. Even though there was a need to increase the role of UX design with implementation professionals, there is still a need for dedicated UX professionals and skills. Good examples of such skills are graphic design, interaction design, and localization expertise. Currently, the UX professionals are seen to operate in an “ivory tower” with insufficient interaction with the rest of the software development team. In addition to improving the UX skills of all the software practitioners, a more agile, iterative, and incremental way of collaboration between dedicated UX designers and software engineers would therefore be a necessity. Svensson and Höst [34] have reported that when adopting agile practices in large organizations, one of the critical issues is to put time and effort into introducing the practice of continuous testing. Our results show that, especially with UX testing, systematic practices for continuous testing were still missing, and the UX professionals faced challenges ensuring continuous UX testing. Many studies of agile development projects state that the adoption of agile ways of working had increased customer involvement and given developers the confidence that they were delivering what the customer needed [13]. In our study, there were difficulties establishing effective and continuous communication and collaboration practices with customers, i.e. internal product programs. Other researchers

have reported that customer involvement practices in agile (in this case, XP) appear to be unsustainable, especially in long and high-pressure projects [13].

The case study focused on the development process. The company would benefit from looking into how design opportunities for UX are fed through the process. Aligning the strategic sensing of the external environment with the operations will give insight into creating dynamic capabilities. Understanding the experience the product is designed for and integrating the goals for UX through the processes will establish decision rules about resource allocation. This will allow the company to focus on areas that are critical to innovation and successful competition.

The fact that the organization in question was in the phase of adopting and introducing new ways of agile working and lean thinking into the organization most probably had an impact on the findings. For example, the unclear role of UX professionals in agile teams and the need to establish procedures to support customer leadership in UX highlight some findings that were still under the process of finding optimal solutions that could be adopted throughout the organization. Other researchers have also found that the experience with agile methods has an impact on both the satisfaction [23] and effectiveness of the work [29]. We therefore assume that the results described here are descriptive for a company undergoing the transition phase, but they can most probably not be generalized for situations in which the agile ways of working have already been established and the transition phase is over. We think that the practical problems illustrated by our findings give hints to researchers, however, where practitioners face problems adopting methods and integrating different methodological approaches into their everyday lives.

VII. CONCLUSION

This paper reports experiences from a case study that explores UX practices in an organization undergoing lean transformation. The case study was conducted within one year of the beginning of the lean transformation process. The results show that the lean transformation and adoption of agile software development practices had been successful in placing UX and UX-related documentation in a central role in planning and goal setting.

Challenges were identified in the following main issues:

- The UX design was not considered very agile. After completion of the main UX design documentation, UX designers had little visibility into how it was implemented and tested in practice. This allowed little opportunity for iteration and collaboration between the UX designer and software engineers.
- Better integration of UX design activities into the software development tasks of scrum teams, including issues such as whether to have a centralized or decentralized UX staff and synchronization of UX work with software development.
- Sustainable practices for enforcing UX leadership of the internal customer for large projects were still not optimal, and there were challenges in the UX leadership.

ACKNOWLEDGEMENTS

The work has been partly funded by TEKES the Finnish Funding Agency for Technology and Innovation through TIVIT Cloud SW SHOK.

REFERENCES

- [1] Armitage, J. 2004. Are Agile Methods Good for Design? *Interactions*, Jan-Feb 2004, pp. 14-23.
- [2] Barney, Jay. 1991. Firm Resources and Sustained Competitive Advantage. *Journal of Management*, 17(1), pp. 99-120.
- [3] Beyer, H., Holtzblatt, K., and Baker, L. 2004. An Agile Customer-Centered Method: Rapid Contextual Design. *Proc of CP/AU*, 2004.
- [4] Blomkvist, S. Towards a Model for Bridging Agile Development and User-Centered Design. *Human-Centered Software Engineering — Integrating Usability in the Software Development Lifecycle*. Human-Computer Interaction Series, 2005, Volume 8, IV, 219-244.
- [5] Carr, N. 2009. *The bid switch: Rewriting the World, from Edison to Google*. Norton, New York.
- [6] Chamberlain, S., Sharp, H., and Maiden, N. Towards a Framework for Integrating Agile Development and User-Centered Design. *Extreme Programming and Agile Processes in Software Engineering*. *Lecture Notes in Computer Science*, 2006, Vol 4044/2006, 143-153.
- [7] Chong Lee, J. and McCrickard, D.S. 2007. Towards Extreme(ly) Usable Software: Exploring Tensions Between Usability and Agile Software Development. *AGILE 2007*, 13-17 Aug. 2007, pp. 59-71.
- [8] Courtney, H., Kirkland, J., and Viguerie, P. 1997. Strategy under uncertainty. *Harvard Business Review*, Nov-Dec, pp. 67-79.
- [9] Dayton, D. and Barnum, C. The Impact of Agile on User-centered Design: Two Surveys Tell the Story. *Technical Communication* 56(3) (August 2009)
- [10] Denzin, N. 1970. *The Research Act. A Theoretical Introduction to Sociological Methods*, Aldine, Chicago, pp. 297-331.
- [11] Detweiler, M. Managing UCD Within Agile Projects. *interactions* 14, 3 (May 2007), 40-42.
- [12] Dikaiakos, M.D., Katsaros, D., Mehra, P., Pallis, G., and Vakali, A. 2009. Cloud Computing: Distributed Internet Computing for IT and Scientific Research. *Internet Computing*, IEEE, 13(3), pp. 10-13.
- [13] Dybå, T. and Dingsøy, T. Empirical studies of agile software development: A systematic review. *Information and Software Technology*. Volume 50, Issues 9-10, August 2008, Pages 833-859
- [14] Ferreira, J., Noble, J., and Biddle, R. Up-Front Interaction Design in Agile Development. in: *Agile Processes in Software Engineering and Extreme Programming*. *Lecture Notes in Computer Science*, 2007.
- [15] Hassenzahl, M. 2003. The Thing and I: Understanding the Relationship Between User and Product. In: M. Blether, C. Overbeeke, A.F., Monk, and P.C. Wright, ed. 2003. *Funology: From Usability to Enjoyment*. Dordrecht: Kluwer Academic Publishers.
- [16] Hassenzahl, M. and Tractinsky, N. 2006. User Experience – A Research Agenda. *Behaviour & Information Technology*, Volume 25, Issue 2, March 2006, pp. 91-97.
- [17] Hassenzahl, M. 2010. *Technology for All the Right Reasons*. Morgan & Claypool, San Rafael.
- [18] Hussain, Z., Milchrahm, H., Shahzad, S., Slany, W., Tscheligi, M., and Wolkerstorfer, P. 2009a. Integration of Extreme Programming and User-Centered Design: Lessons Learned. *XP 2009, LNBP 31*, pp. 174-179.
- [19] Hussain, Z., Slany, W., and Holzinger, A. 2009b. Current State of Agile User-Centered Design: A Survey. *HCI and Usability for e-Inclusion*. *Lecture Notes in Computer Science*, 2009, Volume 5889/2009, 416-427.
- [20] Jokela, T. and Abrahamsson, P. 2004. Usability Assessment of an Extreme Programming Project: Close Co-operation with the Customer Does Not Equal to Good Usability. *PROFES 2004, LNCS 3009*, pp. 393-407.
- [21] Kane, D. Finding a Place for Discount Usability Engineering in Agile Development: Throwing Down the Gauntlet. *Proceedings of the Agile Development Conference*, 2003. ADC 2003.
- [22] Krippendorff, K. 2003. *Content analysis: an introduction to its methodology*.
- [23] Laanti, M. 2010. Agile Transformation Study at Nokia – One Year After. *Proceedings of LESS 2010*, pp. 3-19.
- [24] Lai-Chong Law, E., Roto, V., Hassenzahl, M., Vermeeren, A., and Kort, J. 2009. Understanding, scoping and defining user experience: a survey approach. *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI '09)*, pp. 719-728.
- [25] Leffingwell, D. and Aalto, J.-M. A Lean and Scalable Information Model for the Agile Enterprise. *Whitepaper*. (2009). <http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/982/A-Lean-and-Scalable-Requirements-Information-Model-for-the-Agile-Enterprise.aspx>
- [26] Mannaro, K., Melis, M., and Marchesi, M. 2004. Empirical analysis on the satisfaction of IT employees comparing XP practices with other software development methodologies. *Proc of Extreme Programming and Agile Processes in Software Engineering*.
- [27] McCarthy, J. and Wright, P. Technology as Experience. *interactions* 11, 5 (September 2004), 42-43.
- [28] Memmel, T., Gundelsweiler, F., and Reiterer, H. 2007. Agile Human-Centered Software Engineering. *Proc of the 21st British HCI Group Annual Conference on People and Computer*.
- [29] Middleton, P. 2001. Lean software development: two case studies. *Software Quality Journal*, 9 (4).
- [30] Prahalad, C.K. and Hamel, G. 1990. The Core Competence of the Corporation. *Harvard Business Review*, 68(3), pp. 79-91.
- [31] Prahalad, C.K. and Krishnan, M.S. 2008. *The New Age of Innovation: Driving Co-created Value through Global Networks*. McGraw-Hill, New York.
- [32] Runeson, P., Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empir Software Eng* (2009) 14:131-164. Springer.
- [33] Sohaib, O., Khan, K. 2010. Integrating Usability Engineering and Agile Software Development: A Literature Review. *International Conference on Computer Design and Applications*, IEEE.
- [34] Svensson, H. and Höst, M. 2005. Introducing agile process in a software maintenance and evolution organization. *Proceedings of the 9th European Conference on Software Maintenance and Reengineering (CSMR)*.
- [35] Sy, D. 2007. *Adapting Usability Investigations for Agile User-Centered Design*.
- [36] Teece, D.J., Pisano, G., Shuen, A. 1997. Dynamic capabilities and strategic management. *Strategic Management Journal*, 18(7).
- [37] Teece, D.J. 2007. Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal*, 28, pp. 1319-1350.
- [38] Wellington, C., Briggs, T., and Girard, C. 2005. Comparison of student experiences with plan-driven and agile methodologies. *Proceedings of IEEE Frontiers in Education Conference*.
- [39] Womack, J.P. and Jones, D.T. 2003. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. Free Press.
- [40] Vermeeren, A., Lai-Chong Law, E., Roto, V., Obrist, M., Hoonhout, J., and Väänänen-Vainio-Mattila, K. 2010. User experience evaluation methods: current state and development needs. *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. ACM, pp. 521-530.
- [41] Vredenburg, K., Mao, J-Y, Smith, P., and Carey, T. 2002. A Survey of User-Centered Design Practice. *Proceedings of CHI 2002*. Minneapolis, USA, 20-25 April 2002, pp 471-478.