Teaching Software Developers to Perform UX Tasks

Tina Øvad
Radiometer Medical
Aalborg University
Copenhagen, Denmark
tinap@radiometer.dk

Nis Bornoe
Aalborg University
Aalborg, Denmark
nis@cs.aau.dk

Lars Bo Larsen
Aalborg University
Aalborg, Denmark
lbl@es.aau.dk

Jan Stage
Aalborg University
Aalborg, Denmark
jans@cs.aau.dk

ABSTRACT
Good UX design is becoming important within the industry when developing new products. This entails that UX skills have to be available in the development processes. This paper investigates the opportunities of using software developers as a UX work resource in the day-to-day working practice. This is done via an action research study where the developers were provided with material concerning a modified AB usability test, by training them in performing this type of work, and by using their feedback to improve the method and the material. The overall result of the study is positive and it is found that by using the developers' feedback in the modification process, the method has truly become applicable within an agile, industrial setting. In combination with a guideline and template this has induced the developers to feel confident in independently performing this type of work.

Author Keywords
User Experience (UX) design; Usability; Agile software development; Scrum; Training; Action Research.

ACM Classification Keywords
D.2.2 Design Tools and Techniques: User interfaces; H.5.2 User Interfaces: User-centered design; H.5.2. User Interfaces: Evaluation/methodology; H.1.2 User/Machine Systems; K.6.3 Software Management: Software development.

INTRODUCTION
A company's requirement for UX design introduces demands for UX skills to be available for the development processes. Within the industry an emerging tendency to employ a dedicated UX team in the company can be observed (Øvad and Larsen, 2015a). This corresponds well with stage 4: "Dedicated UX Budget" in the corporate UX maturity model (Nielsen, 2006). In accomplishing this some obstacles have been reported. Not all organizations have the resources needed to hire usability specialists or external consultants (Bruun, 2010). In organizations with both designers and developers cross-disciplinary collaboration can strengthen development, but the act of collaborating has been found challenging (Latzina and Rummel, 2003; Sy, 2007). Some see a UX team as a bottleneck or an organizational block in the development process (Ferreira et al., 2012). Several obstacles has been found when working with usability and UX design in an agile development context such as Scrum (Bornoe and Stage, 2014; da Silva et al., 2011; Ferreira et al., 2012; Lárusdóttir et al., 2013; Salah et al., 2014; Sy, 2007). The contrast between the workload of usability and UX activities, and the simplicity, speed, (Larusdottir et al., 2010), the iterative nature (Ferreira et al., 2012; Sy, 2007), and focus on sprint completion (Bornoe and Stage, 2014; Sy, 2007) in Scrum is challenging. One recommendation is to integrate usability work into day-to-day tasks (Lee and McCrickard, 2007). A specific suggestion is to investigate how usability work can be conducted on a small scale so the activities can be integrated into sprints (Larusdottir et al., 2010). Informal and lightweight approaches are already commonly followed in industry (Bornoe and Stage, 2014; Lárusdóttir et al., 2013). Bruun and Stage (2014) show how basic usability evaluation training is a fast and cheap approach to provide developers with minimum skills. For example, basic training can provide developers with a better understanding of user perspectives, while not replacing usability practitioners (Eriksson et al., 2009). Instead basic training can strengthen the collaboration between non-technical and technical roles (Latzina and Rummel, 2003).

The software industry has adopted the agile development approach and especially the Scrum framework is widely used (Larusdottir et al., 2010; Øvad and Larsen, 2015a). The agile framework states that to be truly agile, every team member should be able to perform every given work task. It therefore seems rational to train software developers in conducting certain usability and UX tasks (Latzina and Rummel, 2003) – and thus make use of the software developers' already present domain knowledge (Bruun et al., 2014; Høegh et al., 2006), and potentially facilitate a shared language within the development team (Ferreira et al., 2012). Especially since an emerging tendency to employ a dedicated UX team in the company can be observed (Øvad and Larsen, 2015a). For companies on level 3 of Nielsen's (2006) corporate UX maturity model ("Skunkworks UX"), this might be the only feasible solution. For level 4 ("Dedicated UX Budget") organisations, this approach facilitates that a dedicated UX team will have more time and resources to focus on e.g. the overall company UX strategy, collecting user requirements, evaluating, etc.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

OzCHI '15 , December 07 - 10 2015, Melbourne, VIC, Australia
Copyright © 2015 ACM 978-1-4503-3673-4/15/12... $15.00
DOI: http://dx.doi.org/10.1145/2838739.2838764

To address these obstacles, we set out to investigate in detail if and how software developers can obtain usability and UX skills and their motivations for doing so. Our aim was to identify potential problem areas and success factors and make recommendations accordingly. In concrete terms, we have selected to modify the comparative usability testing paradigm (here denoted AB-testing). We did this by adopting an action research approach, which makes it possible to address organizational processes by means of having developers take action and improve the process (Baskerville and Wood-Harper, 1996, 1998; Davison et al., 2004; McKay and Marshall, 2001). This study is defined as part of a larger collaborative practice research (Mathiassen, 2002) project, carried out in a company developing medical devices. In the remaining part of the paper we will: Summarize related work; provide an overview of our research method; present our findings; discuss and conclude on our findings.

RELATED WORK

Several approaches about involving software developers in usability work have been proposed.

Frameworks, guidelines, and tools

Different approaches about involving software developers in usability work have been proposed to overcome the described obstacles. Several frameworks, guidelines, and tools have been presented to support different stages of usability and UX work when no or limited expertise is available. For example, it has been proposed to provide guidelines to support the elicitation and specification of usability requirements (Juristo et al., 2007), or to provide a set of HCI methods with information about when and how to apply them in the development process (Ferre et al., 2005). Another example is to provide tools, such as a conceptual tool to support problem identification in a usability evaluation (Skov and Stage, 2005).

Training

We have found that not much research has been conducted in the area of training software developers in usability and UX methods. It has been emphasized that most of existing studies about training developers have been conducted with students as participants (Bruun, 2010). Students may have the same competences as developers in industry, but do not face the same organizational circumstances (Ferreira et al., 2011), obstacles, and settings in which user-centered design (UCD) is performed in industry (Svanæs and Gulliksen, 2008). In this paper we only focus on research about development in a practical context.

Studies have shown positive results, but also report several limitations when training developers. Several studies report increased awareness and understand of usability engineering (Eriksson et al., 2009; Häkli, 2005; Karat and Dayton, 1995; Latzina and Rummel, 2003). Training has been used to establish a corps of usability advocates among developers (Karat and Dayton, 1995) and to increase interest in usability engineering and communicating with usability specialists (Latzina and Rummel, 2003). In one study the developers gained increased knowledge and awareness of the context of use resulting in a more clearly holistic view of the use of the software (Eriksson et al., 2009). Bruun and Stage (2014) trained developers in formative usability evaluation and data analysis, and included the developers in a redesign process (Bruun et al., 2014). As a result the developers were able to identify a significant amount of usability problems compared to usability specialists. Regarding impact, the evaluations resulted in increased downstream utility (Bruun and Stage, 2014). Training has been used to improve the developers' skills in designing and evaluating graphical user interfaces (GUI) (Bornoe et al., 2014; Bruun and Stage, 2014; Häkli, 2005; Latzina and Rummel, 2003; Nielsen et al., 1992). Usability training workshops have been used to make developers more self-supporting when designing a GUI. It is reported that by making abstract usability principles concrete through training, developers were able to apply the principles and design a cohesive GUI (Nielsen et al., 1992), generating potential GUI solutions (Latzina and Rummel, 2003), and correcting usability problems and proposing redesigns (Bornoe et al., 2014). Inclusion of developers has also been beneficial regarding utilization of domain knowledge when correcting usability problems (Bornoe et al., 2014; Bruun et al., 2014; Eriksson et al., 2009).

The studies also report several limitations in the gained skills. Developers find it difficult to conduct evaluations as structured as usability specialists (Häkli, 2005), interpret observations (Eriksson et al., 2009), they lacked clear descriptions of the impact, cause, user actions, and to support the observations with data. Problem descriptions were of a lower quality when compared to usability specialists (Bruun and Stage, 2014), see how findings can be used for radical design changes (Bruun et al., 2014), and keep focus in a design process (Bornoe et al., 2014).

Expertise

An argument about involving developers in usability and UX work is that the developers will be the ones implementing the changes and they often accommodate domain knowledge (Bruun et al., 2014; Høegh et al., 2006). Especially for complex systems, domain knowledge has been empathized as being essential to meet usability goals (Chilana et al., 2010). One study reports that having developers observe usability evaluations provided a richer understanding of usability problems including the severity and use situations they occurred in, and their empathy towards the users. This understanding about the users and work processes was found to inspire future system development (Høegh et al., 2006). Studies have looked into collaboration and active involvement of developers when redesigning user interfaces. It is reported that the developers gained a deeper insight of the identified usability problems and could include domain knowledge not known to usability specialists (Bornoe et al., 2014; Bruun et al., 2014; Høegh et al., 2006). E.g. because of their understanding of the system they can spot minor details that easily can be fixed (Eriksson et al., 2009).

As demonstrated here, a large body of research supports the idea of training developers to perform UX and usability tasks. However, the experiences of doing so also

points towards obstacles and still unsolved problems. Our aim is to further investigate the notion and address some of the reported difficulties.

RESEARCH METHOD AND INITIAL STEPS

In the present study we are following an action research approach. Action research calls for an iterative process involving both the developers and the researchers acting together to define a diagnosis, plan and conduct an action intervention, followed by evaluation and reflection (Davison et al., 2004). The gain of this approach is the opportunity for addressing complex real life problems together with addressing organizational processes by means of having the developers take action and improve their work situation (Baskerville and Wood-Harper, 1996, 1998; Davison et al., 2004; McKay and Marshall, 2001).

The present work is a part of a larger research effort to improve the usability and UX design processes in agile software development. The research is conducted in collaboration with Radiometer Medical and Aalborg University. Radiometer develops medical devices and is therefore under strict regulatory demands of being in compliance with e.g. the usability standard put forward by U.S. Food and Drug Administration (FDA) (ANSI/AAMI HE75:2009, 2009) and ISO (2010).

The project as a whole follows the Collaborative Practice Research approach as put forward by Mathiassen (2002). This approach is an action research methodology, in which it is possible to connect the need to understand the current practices of working with UX, usability and agile software development with the need to integrate these two frameworks to improve the final product. The methodology furthermore offers a structure for the company, allowing the researchers and software developers to collaborate by combining *"action research, experiments, and conventional practice studies to strike a useful balance between relevance and rigour."* (Mathiassen, 2002). For the overall research project an action research protocol was developed, which documented initial thoughts, roles, controls framework, usefulness, documentation, transferability, decisions for each of the action research criteria (Nielsen, 2007). Relevant parts of the protocol are documented in the following section. The encounters important for the present study are included, as well as the roles of the participants and the interview guides. However, we do not have the space here to make a full account.

Initial observations

To initiate the overall research project, R1 (see Table 2 below) was hired to work as a UX designer in the software department. For five months she participated in the daily work and alongside made observations on how Radiometer worked with usability, UX design and Scrum on a day-to-day basis and an in-depth diagnosis was made, which is described in (Øvad, 2014). This diagnosis corresponded with the initial assumptions by Radiometer prior to R1's employment.

Based on the diagnosis of the problem situation, a literature study concerning the work with usability and UX in an agile development environment was performed by R1. Three main findings emerged and are described in (Øvad, 2014). Based on these findings it was suggested to have the software developers do some of the usability and UX work themselves. This solution mitigates the identified problems by achieving two goals:

- The software developers can perform certain, minor usability and UX tasks themselves, thus reducing the UX bottleneck
- A shared language and understanding within the development team

To address the suggested solution, it was agreed between Radiometer and the researchers that suitable usability and UX methods should be selected and adapted to fit into a Scrum process. The modification should also take into account that the performing actors would be software developers, not UX experts. Hence, the overall purpose of the overall research project was agreed upon to be the investigation to what extent a UX toolbox, developed to be used by software developers in an agile software environment can facilitate synchronous work with agile development, usability and UX design.

A prioritised list of usability and UX methods was identified through a literature study (Øvad and Larsen, 2014). The present paper presents the work with AB-testing. A classic AB test is a simple way to test the effect of changing one variable in a design, e.g. the colour of a button, against the current design and determine which design is perceived as the best solution. AB testing is a way to validate that any new change to a variable is improving the product before making the final change.

Participants

Seven software developers participated in the present study. See Table 1 for their profiles. All of them are employed in the software department at Radiometer, but working on two different projects.

ID	*Years of work experience*	*Interested in UX Tasks (expressed before the training session)*
D1	*16*	*Yes*
D2	*9*	*Yes and would conduct this type of work to a certain extent*
D3	*11*	*Yes*
D4	*14*	*Yes, to a certain degree*
D5	*6*	*Yes, to a certain degree*
D6	*3*	*Yes, but would not conduct this type of work himself*
D7	*25*	*Limited interest*

Table 1. Developer profiles.

None of the developers had formal training in usability or UX work. However, all had observed user tests during their employment at Radiometer, but never participated or facilitated one themselves.

Four researchers conducted this study. Their roles are shown in Table 2. All researchers have extensive training and experience in usability and UX work and all participated in the data analysis.

ID	Roles
R1	*Insider action researcher, performing all observations and interviews, performing the modification of the method and the material together with R3*
R2	*Outside action researcher, performing the training*
R3	*Outside action researcher, performing the modification of the method and the material together with R1*
R4	*Outside researcher participating in the interpretation of findings and conclusions*

Table 2. Researcher profiles.

Key encounters

The present study had six key encounters, summarized in Table 3. All encounters took place on-site at Radiometer. The AB-test method was modified in two iterations based on the analysis from encounter 1 and 3. Due to the real-life conditions, not all of D1-7 were available for all encounters, which is reflected in Table 3.

Date	Enc.	Activity	Participants
Dec. 2014	1	Initial Interviews concerning the developers' expectations and reservations	R1, D1-7
Dec. 2014	2	Training session	R2, D1-5
Dec. 2014	3	Interviews collecting training experiences	R1, D1-5
Feb. 2015	4	The developers applied the method on their own.	R1, D1, D2, D4 and D7
Feb. 2015	5	Final interviews	R1, D2, D4 and D7
Feb. 2015	6	Evaluation of the results from encounter 4	In-house UX designer at Radiometer

Table 3. The action research encounters.

We have elected to place the detailed schedule for the training (Encounter 2), as well as examples of the used materials in the subsequent Findings section instead of here.

All interviews were conducted as semi-structured, face-to-face interviews. The duration of the initial interviews (Encounter 1) was between 14 and 34 minutes. The interview guide included the following themes: Background of the participant and years of relevant work experience; work field; knowledge about UX and usability and attitude towards these work fields; the possibility that they will use the method on their own; any prior knowledge concerning the method; final remarks.

The duration of the second interviews (Encounter 3) was between 18 and 31 minutes. The interview guide included the following themes: The participant's output from the training; their overview of the method; things they would like to change concerning the method; things they would like to change concerning the material; remarks for the method and the material; final remarks

The duration of the final interviews (Encounter 5) was between 15 and 47 minutes. The interview guide included the following themes: How the planning, test and analysis were carried out and the logistic concerning this; the results and what they could be used for in the future; remarks for the method and the material; final remarks.

All interview encounters were documented through audio recordings and observation notes. The training session and the session where the developers applied the method on their own were documented via video recordings and notes. Following each encounter, the participating researchers (R1-3) conducted a debriefing.

After having transcribed the interviews, R1 analysed them and the notes by performing a meaning condensation of the data (Patton, 2015), followed by performing five steps in a cyclic manner: compiling, disassembling, reassembling, interpreting and concluding, as described by Yin (2010). This iterative process resulted in the identification of some overall themes. All themes are described in the following, including quotes from the interviewees related to the respective themes. By integrating the analysis into the action research process, we allowed feedback after each iteration. This formed the basis for the iterations and modifications of the method and the material.

Following the evaluation performed by an in-house UX designer at Radiometer on the results from encounter 4, we closed the action research process for this method, since she assessed the method and material's usefulness and found it highly applicable. The materials are described in details in (Øvad and Larsen, 2015b).

FINDINGS

This section presents the findings related to the different encounters.

Encounter 1: Initial Interview

The developers (D1-D7) were interviewed individually prior to the training session. The subsequent analysis revealed three major themes; *expectations towards the particular method and UX activities in general*; *confidence in their capabilities,* and the *usefulness of the work*.

Expectations

The expectations to the method ranged from; "*I expect a lot of work in order to clarify a little thing*" [D6] to "*A way to quickly find a solution*" [D2]. Furthermore, all developers expected to be very clear about the parameters they are to test and that the outcome would be clear, quantitative results presented in some sort of report.

Confidence

All found the idea of doing certain UX tasks interesting. However, the responses clearly indicated they lacked confidence in their abilities. Some would prefer the dedicated UX team did the work. D4 pointed out: "*I would feel more comfortable if there were other people, who are a bit more experienced with this sort of work*" and D5 said: *"I would prefer that our UX team perform*

these tasks". When asked if he could conduct such work if he had some guidelines, D6 responded: *"I do not think I would be any good at facilitating such a test"* [D6]

Usefulness

All of the developers would use the results from an AB-test, but only three of them expressed willingness to conduct an AB-test on their own. D1 said: *"I would definitely use the data from an AB-test. Organise… I would like to participate, but I'm not sure of organizing"*.

D3 pointed out that: "*I sense that by an AB-test, you can try different options and I think it will benefit the product in the end*".

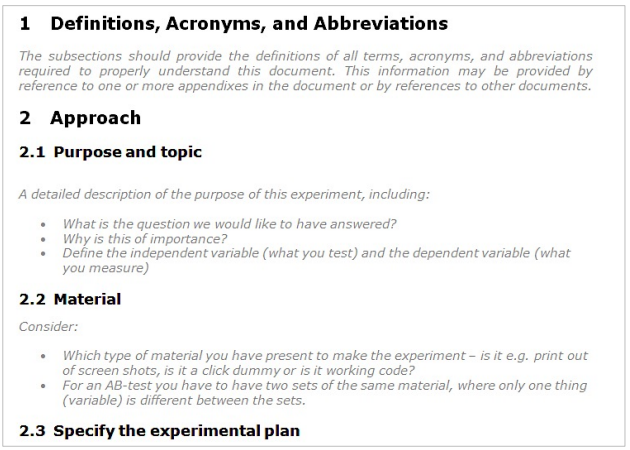
All expected the results would be used in the product development. However, D2 pointed out: *"I can see a problem in that we do not have that much to say (design-wise)"*. D3 expects that: "*… process-wise it will move faster. Right now it seems like a user test is conducted and then months go by before we actually have the experiences and results to use in the development. I hope this method can speed up the process"*.

First iteration of the training process and materials

Based on these findings, a modification of the comparative usability test paradigm (Rubin and Chisnell, 2008) was instituted. We denote the method AB testing. It was in particular the concerns about framing this type of work; difficulties of estimating the time frame; and the company's general needs and policies, which led to this choice. It was clear the generic approach must be tailored to be applicable within one single sprint and the material should be lightweight, but still detailed enough to guide developers without a specialised background in conducting this type of work themselves. Furthermore, since the products of interest are characterised by embedded software running on physical devices, the method must accommodate the physical aspects as well. These considerations led to a modification towards a more qualitative based approach, relying on data from a small sample size.

Guidelines and templates for planning and reporting were developed from these requirements. This is supported by the observation by (Nielsen et al., 1992) that novice practitioners need structure, especially if conducting tests are not their main task.

During the diagnosis phase at Radiometer it was observed that the company made use of different artefacts in the development process. Especially the use of different templates was widely used to document the work, and correctly filled out documents is a requirement, when validating the different products. This template-based approach was adapted to our work, where greyed out text in the template is to guide the developers in performing an AB-test, see Figure 1 (only intended to give a broad impression), for more details see Øvad and Larsen (2015b). In order to further support the developers, a more in-depth guideline concerning AB-testing was developed as well.



1 Definitions, Acronyms, and Abbreviations

The subsections should provide the definitions of all terms, acronyms, and abbreviations required to properly understand this document. This information may be provided by reference to one or more appendixes in the document or by references to other documents.

2 Approach

2.1 Purpose and topic

A detailed description of the purpose of this experiment, including:

- *What is the question we would like to have answered?*
- *Why is this of importance?*
- *Define the independent variable (what you test) and the dependent variable (what you measure)*

2.2 Material

Consider:

- *Which type of material you have present to make the experiment – is it e.g. print out of screen shots, is it a click dummy or is it working code?*
- *For an AB-test you have to have two sets of the same material, where only one thing (variable) is different between the sets.*

2.3 Specify the experimental plan

Figure 1. The first steps of the first iteration of the planning template. The full version of the final template is available from: http://UXToolbox.es.aau.dk

The training session layout was based on principles and designs used by earlier studies, following a "presentations and exercises" approach (Bornoe et al., 2014; Bruun and Stage, 2014; Eriksson et al., 2009; Latzina and Rummel, 2003). By combining short lectures with hands-on exercises we wanted to engage the participants (Nielsen et al., 1992) and have an opportunity to ask questions and getting clarifications. The duration of the training session was set to one day and the agenda is shown in Table 4.

Encounter 2 Training Session

R2 conducted a one-day training (7 hours) session in conducting AB-testing, where five of the developers (D1-D5) participated. The participants were divided into two teams, corresponding to the expected resources available for real-life tests. Each group applied the learned theory on two real life cases taken within the company. One test objective was to determine the placement of a button and the other was to determine the wording of a button. The training activities are shown in Table 4.

Activity	***Details***
Introduction *(20 min.)*	Essentials of AB-testing.
Planning and preparing *(60 min.)*	Clarification of objectives Experimental design (Setting independent and dependent variable) Decide on tasks or actions Practical concerns
Conducting a test *(25 min.)*	Advice and expectation about carrying out an in-the-wild study.
Small break *(15 min.)*	
Hands-on exercise #1: Plan and prepare an AB-test *(60 min.)*	Set up evaluation goals and objectives. Prepare the analyzer.

Lunch *(30 min.)*	
Hands-on exercise #1: Plan and prepare an AB-test continued *(20 min.)*	Set up evaluation goals and objectives. Prepare the analyzer.
Hands-on exercise #2: Conduct a test *(120 min.)*	Find participants and run the AB-test
Small break *(10 min.)*	
Analyze test results and report outcome *(30 min.)*	Qualitative and quantitative data analysis How to report results
Hands-on exercise #3: analyze and report the collected data *(60 min.)*	Decide on which methods to use for analysis and interpret the results
Plenary session *(30 min.)*	Present the analyzed test results.

Table 4. Plan for the training session.

Encounter 3: Collecting Training Experiences (second interview)

The five developers (D1-D5) who participated in the AB-test training were interviewed after the training session. The interviews were performed to capture how the training had affected the developers. The analysis revealed four themes; experiences and issues; confidence and usefulness and training materials.

Experiences and Issues

Overall the developers were satisfied with the training; especially the hands-on exercises were very successful. All experienced some issues and made mistakes. D2 pointed out: "*It was first after the first three participants that we remembered to ask the follow up questions, it was obvious that we became better over time"* and D3 felt that: *"In the beginning we were very awkward and we felt a bit stupid*".

The most important issue was the problem of finding test participants. When asked what they would find the hardest part to do, D1 stated: "*…looking for the test participants, that is a problem… getting the participants – that was really challenging*". Furthermore, all of the developers found the qualitative analysis hard, and it was fairly obvious the developers did not fully grasp the concept of qualitative data analysis.

An important insight was noted by e.g. D1: *"Even if you want to test something very, very small – there is actually a lot of preparation and a lot of after-work after the test*"

Usefulness

All now expressed they would both use data obtained by a test and also conduct an AB-test on their own. D1 expressed this: "*I feel comfortable in any of the steps*"

Training materials

The developers were very positive towards the materials. D3 stated: "*I think they (guideline and templates) were very professional and thorough – and I liked the greyed out guiding texts*". However, some issues with the templates for scoping and reporting the findings were pointed out, e.g. by D4 *"It is just that the test script looks quite intimidating cause there are so many entries, but you just need to understand that some of them are not applicable*".

Confidence

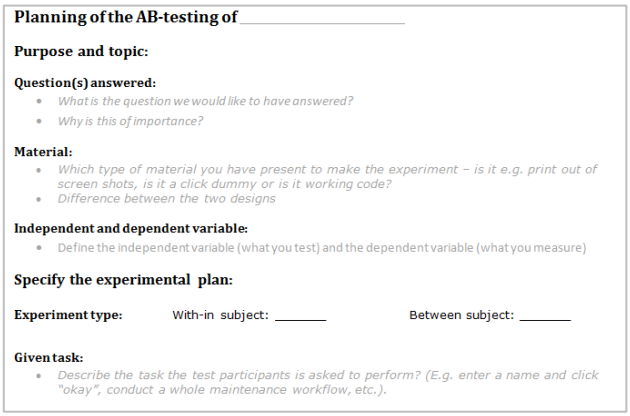
After the training, all felt secure in conducting an AB-test. The most important reason given was they could rely on a structured and established process in combination with the provided materials. D2 pointed out: "*Just to know what making such a test involves and what challenges there are – I think that is healthy*".

From observations during the training and the interviews, the importance of the guidelines and especially the templates was evident, as also noted above. D5 said: *"I think the report should just be a one-pager, where you have five lines to describe the purpose and some check boxes to check concerning if it is a within – or between subject design, etc. – it has to be as easy as possible"*.

Materials should be as lightweight and accessible as possible. Even though it had caused problems; it became clear the main analysis focus should be qualitative, as the scope would be small-sample testing.

Second iteration of the materials

Based on these findings, the templates were modified and made even more accessible and lightweight, and all materials were merged into one single template, thus including both a planning and a report part, Figure 2 shows the first steps of the template (although only broadly illustrated here, due to space limitations). For more details see Øvad and Larsen (2015b). Emphasis was devoted to support the data analysis part. This iteration of the materials was then used in encounter 4.


Planning of the AB-testing of _____

Purpose and topic:

Question(s) answered:
- *What is the question we would like to have answered?*
- *Why is this of importance?*

Material:
- *Which type of material you have present to make the experiment – is it e.g. print out of screen shots, is it a click dummy or is it working code?*
- *Difference between the two designs*

Independent and dependent variable:
- Define the independent variable (what you test) and the dependent variable (what you measure)

Specify the experimental plan:

Experiment type: With-in subject: _____ Between subject: _____

Given task:
- *Describe the task the test participants is asked to perform? (E.g. enter a name and click "okay", conduct a whole maintenance workflow, etc.).*

Figure 2. The first steps of the second iteration of the planning and report template.

Encounter 4: The developers applied the method on their own

Four software developers (D1, D2, D4 and D7) planned, conducted and analysed an AB-test on their own, two months after the training. This work was handled as any other development task in the software team. The task had been defined and assigned story points before being placed in the product backlog. In this sprint the task was moved to the sprint backlog and performed by three of the developers who had participated in the AB-test training and one (D7) only participated in the initial interview. R1 observed the developers when they planned, conducted and analysed the AB-test. The test objective was to determine which of two keyboard layouts should be included in the final product.

The developers decided to split into two teams and perform a test each. They had the responsibility for organising all activities. Details are showed in Table 5

Activity	Time spent	
	Team 1	Team 2
Planning	145 min.	145 min.
Conducting test	75 min.	90 min
Data Analysis	75 min.	80 min

Table 5. Time used by the developers for planning, conducting and analysing an AB-test.

Observations

The developers used the templates extensively to assist the planning of the test and it facilitated some good discussions concerning the different choices to make regarding the test, such as selecting a within- or a between subject test design, how data would be obtained, etc. The extensive reliance on the materials meant the tests were designed almost exactly as the ones they performed during the training session. However, one team developed an extra template for the test.

During the test, the developers handled the test participants well. Both teams videotaped the tests, made notes and recorded the timing. Both teams attempted to conduct a t-test, even though they did not have enough test participants. Generally, the data analysis seemed to be the biggest obstacle and the teams had problems comprehending and applying the qualitative analysis. This observation is supported by Skov and Stage (2005).

Encounter 5: Final Interviews

Similar to the previous encounters, data was captured via interviews. The analysis produced three themes seen previously: Experiences and issues; confidence; and usefulness. A fourth theme emerged, related to the two last ones: Attitude to UX work.

Experiences and Issues

Overall all of the developers were satisfied with the execution of the test and the outcome. D2: "*I think it was excellent and it was obvious that we had tried it once before. … I think it went much better than last time*".

The observation about strict adherence to guides and templates was verified in the interviews and the participants confirmed the level of detail as appropriate.

In relation to the planning and execution of the test, D4 pointed out that: *"I think it was a fairly simple test, and maybe some minor things got too much attention… I think the hard part is coming up with the A and B. After that it should be really straight forward.*" None of the developers experienced problems in finding test participants.

In relation to the observed difficulties with qualitative analysis D7 pointed out: "*There were some exciting comments – and some of them recurred*". However, none of the groups conducted deeper qualitative analysis besides writing down the test participants' comments. D1 noted: *"It would be nice to touch up the analysis part"*.

Usefulness

All developers are sure the results from the test can and will be used within the development process at Radiometer. D1 answered: "*Definitely, definitely! Not only the things we set out to test, but also the things we didn't plan to test (additional finding)*".

Confidence

All of the developers would perform an AB-test again. D7 pointed out: "*You are helped with this material*" and D2 said: "*Yes, but maybe not on my own initiative – it depends on the tasks lying on my desk*".

Attitude towards UX work

The participants expressed an increasingly positive attitude towards UX work during the process. D2 pointed out that: "*It was a nice experience. We are not used to conduct tests ourselves so being able to use our work … I think it is healthy to participate in such stuff.* "

Encounter 6: Evaluation

In a final step to evaluate the results, an in-house UX designer separate to both the teams and researchers was called in and asked to review the outcome of the tests. She confirmed the quality of the obtained information and the value produced for the company: "*The findings are definitely usable. We have obtained a clear indication of the direction to go. One test showed no clear preferences, which in itself is a good thing, as one solution would be much more expensive to implement compared to the other.*" Commenting on the data she observed that both the qualitative and quantitative data was useful, although she would have preferred more observation data. She stated that: "*We have been able to compare this AB-test to earlier tests we have made*".

DISCUSSION OF THE STUDY FINDINGS

The reported action research study demonstrated the feasibility of training developers to perform certain UX tasks in an agile environment. A series of interviews supported by observations was used to provide evidence of how the participants' expectations, confidence and perceived usefulness of the work evolved during the process.

During this research study it was noticeable that the involved developers all displayed an increasingly positive

attitude both toward the presented method, as well as the confidence in their ability towards conducting usability and UX work themselves. The training and application of the method provided new knowledge and skills to the participants as well as an accommodating attitude towards usability and UX work. Concerning the applied training method, it has been successful as indicated by the participants; much of the success is ascribed to the practical approach and the possibility to use real life tasks in the training.

Limitations of developers' skills

Other studies have found that developers find it difficult to interpret observations (Eriksson et al., 2009); to support the observations with data (Bruun and Stage, 2014); and to understand how findings can be used for radical design changes (Bruun et al., 2014). This study confirms and provides further evidence towards this. We saw problems with qualitative data analysis and although the external evaluator found the results useful, she would have preferred more observation data.

Impact of developers performing UX tasks

In line with Larusdottir et al. (2010)'s suggestions, our findings indicate that small-scale developer-driven usability and UX activities can support the understanding between UX designers and developers and indeed provoke an organisational change. Our findings indicate that the developers will be able to enter a dialogue and provide useful input to the UX designers. Furthermore, agile development requires quick and informal evaluation and by this study we have shown that the developers can be trained to perform such tasks.

For companies on level 3 of Nielsen's (2006) corporate UX maturity model involvement of developers can support that UX work is done consistent and systematic. For companies on level 4 the utilisation of developers as a UX workforce can reduce UX as a bottleneck and promote a shared language and understanding between the UX and the software teams, and facilitate that the UX team can focus on tasks requiring expert knowledge.

CONCLUSION

Our aim is to provide software developers with capabilities to perform usability and UX work on their own and thus facilitate a merge of UX work and agile software development. With this in mind we chose to focus on using the developers as a UX work resource in a day-to-day working practice by providing them with material concerning modified AB usability test, by training them in performing this type of work, and by using their feedback to improve the method and the material.

With this action research study we have contributed with empirical knowledge on how to train software developers to conduct minor AB usability tests. Furthermore, we have contributed to the limited research on how software developers can be a part of the on-going work with usability and UX design within companies.

We conclude that the study has successfully demonstrated the feasibility of training software developers to carry out certain usability and UX tasks. A key element of the approach has been to first adapt the method to fit into an agile environment and to provide a set of detailed templates and forms to guide the participants, both elements have been performed via an iterative process in collaboration with the software developers, see Øvad and Larsen (2015b). This proved to be effective and allowed the participating software developers to plan, facilitate and implement a test session on their own within a sprint. In the present study the quality of this work has been evaluated and found usable by an in-house UX designer.

However, it is important to note that we do not see the results of our study as general step towards removing UX specialists from the development process. Rather we regard it as an approach for supporting developers during ongoing day-to-day product development, by performing certain limited usability and UX tasks themselves.

LIMITATIONS AND FUTURE WORK

It is still too early to say if the training of the developers has facilitated an organizational process change at Radiometer. However, as presented in this paper, the developers are indeed capable and motivated for this type of work and this is supported by management. Future work will therefore be to keep track of how often the developers perform an AB usability test on their own. Radiometer management is currently developing Key Performance Indicators (KPI) and different metrics in order to measure the usability and UX work performed by the UX team. It is under consideration to develop KPI's for the usability and UX work performed by the software team as well to fully integrate this type of work in the development process.

The AB usability test method applied in the present study was carefully screened and evaluated as feasible for integrating into an agile environment and teaching non-UX professionals (Øvad and Larsen, 2014). Only a limited number of methods can be expected to fit these criteria. One important obstacle was observed, as the developers showed difficulties with analysing the collected data. This finding will potentially limit the scope of the usability and UX methods suitable for this approach. Next steps are to extend the overall action research study to include other usability and UX methods and a broader base of companies for further evaluation and validation. It should be kept in mind that this type of test cannot secure compliance with the standards put forward by e.g. the FDA, but be an instrument to enhance the quality of the product and ease the compliance work in the long run.

ACKNOWLEDGMENTS

We thank all participating staff at Radiometer Medical. Furthermore, we thank Aalborg University, Radiometer Medical and the Danish Ministry for Science and Education for funding the research presented here. The developed materials and templates can be freely used and are available from: http://UXToolbox.es.aau.dk

REFERENCES

ANSI/AAMI HE75:2009. Human factors engineering-Design of medical devices. Association for the Advancement of Medical Instrumentation (2009)

Baskerville, R.L., Wood-Harper, A.T. A critical perspective on action research as a method for information systems research. J. Inf. Technol. 11 (1996), 235–246.

Baskerville, R., Wood-Harper, A.T. Diversity in information systems action research methods. Eur. J. Inf. Syst. 7 (1998), 90–107.

Bornoe, N., Billestrup, J., Andersen, J.L., Stage, J., Bruun, A. Redesign Workshop: Involving Software Developers Actively in Usability Engineering. In proc. NordiCHI '14. ACM (2014), 1113–1118.

Bornoe, N., Stage, J. Usability Engineering in the Wild: How do Practitioners Integrate Usability Engineering in Software Development?, In proc. HCSE '14 (2014), Springer, 199–216.

Bruun, A. Training software developers in usability engineering: a literature review. In proc. NordiCHI '10, ACM (2010), 82–91.

Bruun, A., Jensen, J.J., Skov, M., Stage, J. Active Collaborative Learning: Supporting Software Developers in Creating Redesign Proposals. In proc.HCSE '14 (2014), Springer, 1–18.

Bruun, A., Stage, J. Barefoot usability evaluations. Behav. Inf. Technol. (2014), 1148–1167.

Chilana, P.K., Wobbrock, J.O., Ko, A.J. Understanding Usability Practices in Complex Domains. In proc. CHI '10, ACM (2010), 2337–2346.

Da Silva, T., Martin, A., Maurer, F., Silveira, M. User-Centered Design and Agile Methods: A Systematic Review. In proc. AGILE2011 (2011), 77–86.

Davison, R., Martinsons, M.G., Kock, N. Principles of canonical action research. Inf. Syst. J. 14, (2004), 65–86.

Eriksson, E., Cajander, Å., Gulliksen, J. Hello World! – Experiencing Usability Methods without Usability Expertise. In proc. INTERACT '09, Springer (2009), 550–565.

Ferreira, J., Sharp, H., Robinson, H. Agile development and user experience design integration as an ongoing achievement in practice. In proc. AGILE2012. IEEE (2012), 11–20.

Ferreira, J., Sharp, H., Robinson, H. User experience design and agile development: managing cooperation through articulation work. Softw. Pract. Exp. 41 (2011), 963–974.

Ferre, X., Juristo, N., Moreno, A.M. Framework for integrating usability practices into the software process. Product Focused Software Process Improvement, Springer (2005), 202–215.

Häkli, A. Introducing user-centered design in a small-size software development organization. Hels. Univ. Technol. Hels. (2005).

Høegh, R.T., Nielsen, C.M., Overgaard, M., Pedersen, M.B., Stage, J. The Impact of Usability Reports and User Test Observations on Developers' Understanding of Usability Data: An Exploratory Study. Int. J. Hum.-Comput. Interact. 21 (2006), 173–196.

ISO. Ergonomics of human system interaction. 9241-210 (2010)

Juristo, N., Moreno, A.M., Sanchez-Segura, M.-I. Guidelines for eliciting usability functionalities. Softw. Eng. IEEE Trans. On 33 (2007), 744–758.

Karat, J., Dayton, T. Practical Education for Improving Software Usability. In proc. CHI '95. ACM, (1995), 162–169.

Larusdottir, M., Bjarnadottir, E., Gulliksen, J. The Focus on Usability in Testing Practices in Industry. In proc. IFIP2010, Springer (2010), 98–109.

Lárusdóttir, M., Cajander, Å., Gulliksen, J. Informal feedback rather than performance measurements – user-centred evaluation in Scrum projects. Behav. Inf. Technol. (2013)

Latzina, M., Rummel, B., 2003. Soft(ware) skills in context: Corporate usability training aiming at cross-disciplinary collaboration. In proc. CSEE&T 2003, (2003), 52–57.

Lee, J.C., McCrickard, D.S. Towards Extreme(ly) Usable Software: Exploring Tensions Between Usability and Agile Software Development. Proc. in AGILE2007 (2007), 59–71.

Mathiassen, L. Collaborative practice research. Inf. Technol. People 15 (2002), 321–345.

McKay, J., Marshall, P. The dual imperatives of action research. Inf. Technol. People 14 (2001), 46–59.

Nielsen, J. Corporate UX Maturity. Nielsen Norman Group, URL: http://www.nngroup.com/articles/usability-maturity-stages-1-4/ (2006), accessed 2.1.15.

Nielsen, J., Bush, R.M., Dayton, T., Mond, N.E., Muller, M.J., Root, R.W. Teaching Experienced Developers to Design Graphical User Interfaces. In proc. CHI '92 (1992), 557–564.

Nielsen, P., 2007. IS action research and its criteria. Inf. Syst. Action Res. (2007), 355–375.

Øvad, T. Agile User Experience. In proc. IHCI '14, IADIS (2014), 397–401.

Øvad, T., Larsen, L.B. The Prevalence of UX Design in Agile Development Processes in Industry. In proc. AGILE2015 (2015a), IEEE, 40–49.

Øvad, T., Larsen, L.B. Developers Love Their Templates – or How to Train Software Developers to Perform UX Tasks. In: Integrating User Centered Design in Agile Development, Springer (2015b).

Øvad, T., Larsen, L.B. Experiences from Training Agile Software Developers in Focused Workshops. In proc. IHCI 2014, IADIS (2014), 355–359.

Patton, M.Q. Qualitative Research & Evaluation Methods, 4th ed. Sage Publications (2015).

Rubin, J., Chisnell, D. Handbook of usability testing: how to plan, design and conduct effective tests. John Wiley & Sons (2008).

Salah, D., Paige, R.F., Cairns, P. A Systematic Literature Review for Agile Development Processes and User Centred Design Integration. In proc. EASE '14, ACM, (2014), 5:1–5:10.

Skov, M.B., Stage, J. Supporting Problem Identification in Usability Evaluations. In proc. OZCHI '05 (2005), 1–9.

Svanæs, D., Gulliksen, J. Understanding the Context of Design: Towards Tactical User Centered Design. In proc. NordiCHI '08 (2008), ACM, 353–362.

Sy, D. Adapting usability investigations for agile user-centered design. J. Usability Stud. 2 (2007), 112–132.

Yin, R.K. Qualitative research from start to finish. Guilford Press (2010).