Taylor & Francis
Taylor & Francis Group

Behaviour & Information Technology

ISSN: 0144-929X (Print) 1362-3001 (Online) Journal homepage: https://www.tandfonline.com/loi/tbit20

How to reduce the UX bottleneck – train your software developers

Tina Øvad & Lars Bo Larsen

To cite this article: Tina Øvad & Lars Bo Larsen (2016) How to reduce the UX bottleneck – train your software developers, Behaviour & Information Technology, 35:12, 1080-1090, DOI: 10.1080/0144929X.2016.1225818

To link to this article: https://doi.org/10.1080/0144929X.2016.1225818

Published online: 08 Sep 2016.

Submit your article to this journal

Article views: 472

View related articles

View Crossmark data

Citing articles: 2 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tbit20

BEHAVIOUR & INFORMATION TECHNOLOGY, 2016
VOL. 35, NO. 12, 1080–1090
http://dx.doi.org/10.1080/0144929X.2016.1225818

Taylor & Francis
Taylor & Francis Group

How to reduce the UX bottleneck – train your software developers

Tina Øvad[a,b] and Lars Bo Larsen[b]

[a]Radiometer Medical, Copenhagen, Denmark; [b]Department of Electronic Systems, Aalborg University, Aalborg, Denmark

ABSTRACT
Usability and user experience (UX) methods come from academic environments, where industrial conditions such as time and resources are not of prime importance. Furthermore, usability and UX methods originate from a time when almost all software developments followed a traditional approach, such as the waterfall model. These two facts entail that existing methods often are too resource demanding and complex to apply directly into today's agile, industrial environments. In this paper we make the claim that methods must be updated and tailored in order to be applicable within the agile, industrial development framework of today. We pursue a solution to simplify well-known methods and to train software developers to perform the UX work. To do this, three methods are modified via an iterative process together with the development of supporting materials. Software developers in three companies are trained in the methods to assess the approach. We find that it indeed is feasible to update and tailor existing usability and UX methods to fit into an agile, industrial environment. Furthermore, we show that it is possible to train developers to perform the usability and UX methods via one-day, *in-situ* sessions using an 'instructor'-teaching approach. The training is based on hands-on exercises and real-life tasks. This further boosts the developers' confidence in performing UX work and promises a better consideration of UX in the development phases. We evaluate our approach through observations of the developers performing the UX tasks on their own at a later point in time.

ARTICLE HISTORY
Received 25 January 2016
Accepted 6 August 2016

KEYWORDS
Training; UX design; usability; agile software development

1. Introduction

User experience (UX) design has matured in recent years within industry and has started to become a competitive factor in product development (Øvad and Larsen 2015).

However, two emerging problems occur when employing usability and UX methods in industry. The first problem is that many methods originate from an academic environment, and thus have not taken real-life conditions in industrial settings into account, especially constraints in time and resources. The second problem is that most usability and UX methods originate from a time when almost all software developments followed the waterfall model. These two facts entail that methods are too resource demanding in terms of time and manpower and difficult to apply into today's agile, industrial environments.

Here we make the claim that methods must be updated and tailored to be applicable within today's agile, industrial development framework. The ability to perform usability and UX work in an agile framework induces different benefits: firstly, the classical agile benefits: transparency, inspection and adaptation (Sutherland and Schwaber 2011); secondly, this supports the potential to be truly user-centred, due to the ability to apply user feedback to every development sprint.

In this paper we pursue a solution, where we simplify well-known methods and train software developers to perform UX work. By doing this, we can facilitate a permeation of UX throughout the whole development process and encourage more transparency within UX work, facilitate a shared language in the development team and minimise UX bottlenecks. By enabling the developers to perform certain UX tasks, we can potentially free resources for the UX specialists in the organisation. In other cases, it will enable companies with no dedicated UX specialists to perform these tasks in-house. The objective for the present work is, therefore, to investigate if and how software developers can be trained to apply UX methods successfully.

We start by looking into related work concerning training of software developers. Next, we present the research methodology. This is followed by a section describing our experiences from adapting and training three well-known methods: Focused Workshop, AB test and Contextual Interview. Finally, we present the discussion and conclusion of the paper.

CONTACT Tina Øvad tina@oevad.com

© 2016 Informa UK Limited, trading as Taylor & Francis Group

2. Related work

The notion of training non-experts to carry out UX tasks has been addressed in a number of studies. These have shown that they can indeed gain an increased awareness and understanding of usability engineering (Eriksson, Cajander, and Gulliksen 2009; Häkli 2005; Karat and Dayton 1995a; Latzina and Rummel 2003; Øvad et al. 2015; Øvad and Larsen 2014).

However, it should be noticed that most of this research used students as experimental participants (Bruun 2010). We view this as a problem, since the students' overall attitude, goals, context, etc., are different from that of those working in real-life environments and they do not face the same organisational circumstances as developers working in the industry. The organisational context is judged to be of great importance when evaluating the outcome of the training and we will, therefore, focus this work on developers in the industry. In the following, we discuss our studies in which software developers from industry have been trained in usability and the UX method.

The first study is by Nielsen et al. (1992). In a series of five workshops, 27 developers were trained to design graphical user interfaces (GUIs). Each workshop was designed as a one-day training session and focused on design principles and guidelines for GUIs and paper mock-ups. The developers worked in teams and used the methods for their own GUI designs.

A follow-up evaluation was made by the authors seven months after the training. Here they looked at one of the participating teams, who had built a complete GUI prototype for one of their products. Heuristic evaluation was used to inspect the prototype, and it not only revealed several usability problems, but also showed that the developers had been able to apply the methods and design a coherent GUI (Nielsen et al. 1992).

The second study is by Karat and Dayton (1995b). Karat and Dayton report two different training programmes.

Firstly, they conducted a six-day on-site training session at an IBM software development lab. The aim was to establish a corps of usability advocates among the developers. This was done by introducing them to Contextual Inquiry, customer interviews, design constraints and standards. The training led to developers conducting numerous inquiries and establishing effective communication between different parts of the organisation.

Both developers and management expressed satisfaction with the training and outcome – a general increase in usability awareness in the organisation.

Secondly, they conducted a three-day workshop, which consisted of 10% formal lectures and 90% hands-on work on a product under development. The developers were seated in small groups of six–seven people, including at least one real end user in each group. They were introduced to a participatory design framework and the focus was on task analysis and paper mock-ups. Karat and Dayton argue that the developers would have an easy job explaining the methods to other co-workers within the organisation and easily be able to apply the methods from the workshop in future projects, because the learning environment was very similar to a real software development environment. However, the authors do not present any evidence for these positive benefits of the workshop's outcome (Karat and Dayton 1995b).

The third study is by Latzina and Rummel (2003). A number of developers from the SAP company were trained at a time in a series of two-day training workshops. Even though SAP had their own usability experts, the company experienced a need to ease the communication between the developers and the usability experts and make the developers more self-supporting during the user interface implementation.

The training sessions were focused on the introduction to Personas, together with a simulation game introducing user scenarios, paper prototyping, style guides and user testing (usability evaluation).

The participants filled out a questionnaire after each workshop. A qualitative analysis of their answers revealed several aspects confirming the developers' increased interest in user-centred design. They were able to develop own ideas for potential solutions and were motivated to engage in more usability training. Furthermore, the developers reported some requirements for organisational changes necessary to fit the learned methods into their development process. They experienced a need for good communication both between team members and collaborating teams in order to succeed with the user-driven approach. Finally, the use of usability standards and user/task information required sufficient time and solid project management (Latzina and Rummel 2003).

The forth study is by Bruun and Stage (2014). They trained eight software developers in a traditional task-based usability test including video analysis in a two-day training session. This enabled the developers to identify 48% of all usability problems in a test case – in comparison to human computer interaction specialists, who identified 62%. Next, they focused on the Instant Data Analysis (IDA) (Kjeldskov, Skov, and Stage 2004) usability test method. This study showed comparable results between developers and specialists. A later follow-up study revealed that developers had fixed approximately 60% of the identified issues, hence accepting the usability corrections as a task in their development project (Bruun and Stage 2014).

These studies show that it is indeed possible to train software developers to successfully perform UX tasks. However, the studies are spread out in time and all report on a single case and method in a particular company. Thus, the extent to which the results can readily be generalised is not obvious. Furthermore, none of the presented research has been conducted within an agile environment with the constraints this entails. In this work, we set out to investigate the effects of training developers in UX methods within an agile development environment across different tasks, methods and in different companies.

3. Research methodology

This section presents the research setting and methodology. We refer to ‘a study’ as the training and evaluation of one usability/UX method within one company. In total six studies are presented.

3.1. The study sites

The six studies are carried out in three different companies. These are briefly described and their motivation for training software developers is given.

Radiometer Medical (*RMED*): RMED is a provider of solutions for acute care testing and develops medical devices. RMED has about 2400 employees worldwide with 250 in R&D and the company headquarter is in Denmark (Radiometer Medical ApS 2015). The US Food and Drug Administration (FDA) is demanding increasingly strict compliance with usability standards such as ISO 9241-210 (DIS 2009). Therefore, RMED has focused on UX for a number of years and employs a dedicated UX team. However, the RMED intents software developers to perform minor UX tasks on their own to minimise potential bottlenecks and to develop a shared language between the UX and development teams. RMED has used Scrum for five years as the primary development framework and employs three-week sprints. (Øvad and Larsen 2016).

TC Electronic (*TC*) produces audio equipment primarily for the music industry, for example, guitar and bass amplifiers, guitar pedals, and sound and picture production – as well as PA systems. TC is a global company with headquarters and main R&D facility in Denmark. Worldwide, TC has about 300 employees, with 30 in the R&D department (TC Electronic 2015).

TC used Scrum as the primary development framework for four years, using three-week sprints. Each development team is a mix of software, hardware and mechanical engineers with a joined product focus. The company has no dedicated UX team or employees, and intends the R&D teams to carry out UX tasks (Øvad and Larsen 2016).

SenDx Medical (*SenDx*) is an America-based company located in California. SenDx develops medical devices and is a subsidiary to RMED. Hence, SenDx is under the same regulatory demands as RMED concerning the usability standards put forward by the FDA (SenDx 2015). SenDx is using Scrum as the development framework and is employing three-week sprints. SenDx has people working with usability, but relies on RMED’s UX team concerning major UX tasks. By upgrading the developers’ skills at SenDx, they will be able to perform certain tasks on site instead of relying on the RMED UX team (Øvad and Larsen 2016).

3.2. Action research

One of the authors worked at both RMED and SenDx during the studies, and she was therefore able to enter into longitudinal studies. This, together with the character of the research, made us choose to frame the study as an action research study. The overall research presented in this paper, therefore, follows an action research approach. The Collaborative Practice Research approach as put forward by Mathiassen (2002) is applied. This has made it possible to connect the needs for understanding the current practices of working with UX, usability and agile software development in the companies, with the need to integrate these two frameworks to improve a final product. Furthermore, this approach offered structure for the company by allowing the authors and software developers to collaborate by combining action research, experiments and more traditional research approaches (Mathiassen 2002). The study performed at TC should be regarded as an initial explorative study and not an action research study.

3.3. Selection of UX methods for the studies

The selection of suitable methods for the studies was partly based on their complexity and partly on the needs and requirements from the collaborating companies (shown in parentheses).

The methods should:

- Enable the developers to perform limited formative testing (RMED and SenDx);
- Enable the developers to get to know the end users (RMED);
- Feed directly into the development process (RMED and SenDx);
- Provide a simple way to gather insights of user behaviour (RMED, SenDx and TC);

- Fit into the companies' agile development process (RMED, SenDx and TC);
- Furthermore, the methods should be fairly easy to learn, plan, conduct and analyse, since non-experts were to perform them (RMED, SenDx and TC);
- Training should be conducted within one day (RMED and SenDx).

These criteria led to a short list:

- Situated observations and interviews, for example, Contextual Inquiry (Beyer and Holtzblatt 1997; Holtzblatt, Wendell, and Wood 2005);
- AB testing (Rubin and Chisnell 2008);
- Different lightweight methods such as:
  - IDA (Kjeldskov, Skov, and Stage 2004)
  - Rapid Iterative Testing and Evaluation (RITE) (Medlock et al. 2002);
- Heuristic Evaluation (Nielsen and Molich 1990);
- Cognitive walkthrough (Polson et al. 1992);
- Think-aloud test (Lewis 1982);
- Focus groups (Krueger and Casey 2002).

Three methods were selected in consultation with the companies. These are:

- Focus group technique, modified by Øvad and Larsen (2014). This is denoted Focused Workshop.
- Comparative usability testing, modified by Øvad et al. (2015). This is denoted AB testing.
- Contextual Inquiry as described by Beyer and Holtzblatt (1997) and Holtzblatt, Wendell, and Wood (2005) and modified in this paper. This is denoted Contextual Interview.

The following sections present the experiences with these three methods. The initial work with Contextual Inquiry is presented in Section 4.1, Focused Workshop is presented in Section 4.2, AB testing in Section 5 and Contextual Interview in Section 6.

3.4. The iterative process

The three UX methods were modified to make them more applicable in an industrial, agile development environment. This was done in an iterative process, where supporting materials were also developed.

The basic process comprised the following:

- Initial interviews with the developers concerning their expectations and wishes;
- Modification of the UX method and development of supporting materials;
- Training session;
- Post-training interviews;
- Modification of supporting materials;
- Developers applied the method;
- Validation of data gathered by the developers;
- Final interviews.

Table 1. Overview of the training and evaluation iterations.

	TC	RMED	SenDx
Focused Workshop		Summer 2014 Summer 2014	
AB testing		Fall 2014 Winter 2015	Spring 2015 Summer 2015
Contextual Interview	Fall 2013 Winter 2014	Spring 2015 Fall 2015	Summer 2015 Summer 2015

Every method ran through this process at least once, where the training materials were updated once between the training session and the evaluation. This process is applied to all methods. Table 1 shows that Focused Workshop has been iterated once, AB testing twice and Contextual Interview three times. For more details concerning the modified methods and the developed materials, see Øvad et al. (2015) and Øvad and Larsen (2014, 2016).

The data collection was triangulated using observations, semi-structured interviews and analysis of the documents created by the developers. This was done during training sessions and when they applied the methods.

Table 1 shows an overview of the training and evaluation iterations.

Our primary focus is the training sessions. However, we will also address how the planning, execution and analysis from the different methods fit into a development sprint without the development team losing too much velocity. For details about the materials, see Øvad and Larsen (2016).

3.5. Participants

In total 28 developers from the three companies participated in the training. All have a background as software developers for embedded devices and have this as their primary task. Only one developer had any formal training in usability or UX work. However, some have observed user tests during their employment, but never participated or facilitated this type of work themselves.

Two researchers conducted this study: one as an insider action researcher, performing all observations and interviews, performing the training in Focused Workshop and the statistical part of the training in AB test at RMED, together with all training at SenDx. The second was an outsider action researcher performing training in Contextual Interview at RMED and Contextual Inquiry at TC. Both have extensive training and experience in usability and UX work, and both participated in the data analysis and the modification of the methods and the materials. Furthermore, two other researchers conducted the training in Contextual Inquiry at TC and AB testing at RMED.

3.6. Research approach

All interviews were conducted as semi-structured, face-to-face interviews and they followed different interview guides according to the purpose. All interview encounters were documented through audio recordings and notes. The training sessions and the session where the developers applied the methods were documented via video recordings and/or notes. Following each encounter, the participating researchers conducted a debriefing and transcribed the interviews. The interviews and notes were analysed by performing a meaning condensation of the data (Patton 1990), followed by five phases, carried out in a cyclic manner: compiling, disassembling, reassembling, interpreting and concluding, as described by Yin (2010). By integrating the analysis into the action research process, we allowed feedback in relation to each iteration and thereby to form the basis for the modifications of the methods and the materials.

3.7. Training objectives

Four training objectives were developed together with RMED management to help evaluate the training outcome and how well the developers mastered the tasks defined in the objectives. We made use of Bloom's revised Taxonomy Table as presented by Krathwohl (2002). The four training objectives were as follows:

(1) The developer should be able to remember and understand the terminology used when performing the given method.
(2) The developer should be able to judge in which cases the method can be applied. Furthermore, the developer should be able to create a plan for the execution of the method.
(3) The developer should be able to apply the method to solve a real-life task, together with the ability to analyse the results obtained from this application.
(4) The developer should be able to evaluate the results and the usefulness of the obtained data and have the ability to use the results to suggest solutions for further development within the given project.

3.8. Training approaches

Dreyfus and Dreyfus (1980) suggest two options, when acquiring a new skill: you can either pick it up by imitation and floundering trial-and-error, or you can seek the help from an instructor or a manual (Dreyfus and Dreyfus 1980).

To identify the best solution, we tried them both. This is presented in the section below in the initial work with Contextual Inquiry and Focused Workshop.

4. Initial studies – Contextual Inquiry and Focused Workshop

We applied both the 'instructor' and 'observe and learn' approach. This was done in two studies. The training sessions were designed to fit into a one-day on-site schedule. This was based on wishes from the companies and to ease the logistics. This is also inspired by the studies discussed in Section 2, where all except one relied on one- or two-day training sessions.

4.1. Contextual Inquiry training at TC ('instructor' approach)

We trained seven developers in Rapid Contextual Inquiry in a one-day workshop (seven hours). The training included the tailored qualitative data analysis methods, as described by Beyer and Holtzblatt (1997) and Holtzblatt, Wendell, and Wood (2005). Training materials consisted of Holtzblatt, Wendell, and Wood (2005), including samples of affinity diagrams, artefact models, etc., associated with Contextual Inquiry. The developers applied the techniques on example cases, such as analysing a video recording of a person making coffee and assembling and testing a hi-fi set.

4.1.1. Results from the Contextual Inquiry training at TC

Interviews with three of the developers revealed that the training had been well received. In particular, they highlighted the practical approach, using small exercises and concrete examples, as very beneficial for their learning outcome. A few weeks after the training session, TC was able to scale the extent of the Contextual Inquiry to fit their sprint rhythm (three weeks) and carry out a Contextual Inquiry as part of the definition phase for a new product. The Contextual Inquiry fit well with their development process and the gathered user insights provided the developers with ideas for future product features.

Even though the interviewees were positive towards the method and regarded it as a good match for their needs, some problems were verbalised. They found certain parts of the data analysis too difficult – especially the creation of the affinity diagram. Consequently, one of the researchers assisted with this. Furthermore, they found it time consuming. The total consumption per developer is listed in Table 2. Except for the presentation, the Contextual Inquiry was performed by three developers.

The Contextual Inquiry as performed here can fit into a single sprint without problems.

The training was regarded as successful in the sense that TC later reported that a number of anticipated features had been dropped due the Contextual Inquiry and the total development time for the product had been reduced by three months.

4.2. Focused Workshop training at RMED ('observe and learn')

Two developers observed and acted as note-takers, while one of the authors conducted a Focused Workshop at RMED. The training included a structured and rigorous guideline to support the developers, together with two templates – one for planning the session and one for reporting the findings. For more details, see Øvad and Larsen (2016). The topic of the workshop was the serviceability of a new product to be launched primo 2016. The workshop participants were service technicians and engineers and an employee from marketing. The final part of the training consisted of analysing the workshop notes, writing a small report and presenting the results.

4.2.1. Results from the Focused Workshop training at RMED

Interviews revealed that the developers had achieved a high degree of confidence in their abilities to conduct a Focused Workshop. Furthermore, they expressed satisfaction regarding the obtained information and insights into the tasks of the participants. They both felt supported by the guideline as a reference.

One of the developers subsequently planned, conducted and analysed a Focused Workshop independently. The time consumption is shown Table 3.

Table 2. Time consumption per participant for the Contextual Inquiry performed by the TC participants.

Activity	Time spent
Preparations for CI	3 hours
Conducting CI	4 hours
Data analysis of CI	14 hours
Presentation of result to team	2 hours
Total	*23 hours*

Note: CI, contextual interview.

The findings indicate that planning, carrying out, analysing and presenting the data from the Focused Workshop can be done in approximately two days. However, we observed that some important issues were left out of the report.

4.3. Discussion and conclusion on the 'instructor' and 'observe and learn' training approaches

Both approaches were well received, especially due to the practical aspect of using hands-on exercises and real-life tasks. However, we observed some issues: The interviews conducted at TC revealed a problem concerning the analysis and it was noted it was too time consuming and difficult for the developers. The RMED interviews revealed that the guideline and templates were beneficial, but important issues, which would have been emphasised in a more structured training session, were omitted from the analysis.

Looking at the training objectives, it was clear that both training approaches fulfilled the first two objectives. The 'instructor' approach also met objectives three and four. The 'observe and learn' approach was not successful in either objective three or four. These concerned the ability to analyse and evaluate the results and suggest solutions for further development. This experience indicates that the 'observe and learn' approach does not provide sufficient structure and guidance. Even though this approach is well received and infuses the developers with a high level of confidence, it introduces too much randomness in the learning process and is therefore discarded. Hence, we will apply the Instructor training paradigm in the following iterations.

As a result of these initial studies, it was decided to continue with training sessions as one-day hands-on training courses carried out on-site with real-life tasks. This is supplemented by guidelines and templates.

5. AB testing training

Training in the AB test method is structured according to these findings (see Øvad et al. 2015; Øvad and Larsen 2016).

Table 3. Time consumption per participant for the Focused Workshop.

Activity	Time spent
Planning the workshop	6 hours
Workshop	1.5 hours
Analyse notes	5 hours
Presentation (incl. preparation)	2 hours
Total	*14.5 hours*

The training was performed at both RMED and SenDx. In the following sections, we first describe the training and findings from RMED, followed by SenDx.

5.1. AB test at RMED

Five participants were introduced to AB testing through a one-day training session at the company facilities. The participants were divided into two teams. Each team applied the method on real-life cases extracted from the sprint backlog. One test objective was to determine the optimal placement of a button and the other was to determine the best label for a button. Both tasks related to the design of the GUI for RMED's next-generation blood gas analyser (see Øvad and Larsen, 2016). For a detailed training plan, see Øvad et al. (2015).

All participants were interviewed after the training session. Overall, they were satisfied with the training. We saw once again that hands-on exercises and real-life tasks were popular. They expressed a strong preference for the structured approach and the support materials. Similarly to the CI, we observed problems with the qualitative analysis (Øvad et al. 2015).

Two months after the training session, four software developers planned, conducted and analysed two AB tests. The objectives were to determine which of two keyboard layouts should be included in a final product GUI. Each team conducted an AB test (Øvad et al. 2015). The timing is shown in Table 4.

We observed that the developers made extensive use of the provided templates when planning the tests, which facilitated good discussions concerning the experimental design and the proper use of terminology (Øvad et al. 2015; Øvad and Larsen 2016). The developers generally performed the tests successfully. Some problems were again observed concerning the data analysis, especially concerning comprehending and applying the qualitative analysis. Interviews supported this and a participant stated: 'It would be nice to touch up the analysis part' (Øvad et al. 2015). The potential of gathering qualitative data was something all developers were aware of. However, they did not perform a more in-depth extensive analysis. Instead, test subjects' comments were noted down and used directly. The post-test interviews revealed that all developers were surprised by the amount of useful data they obtained and they had used the support materials extensively (Øvad et al. 2015).

An in-house UX designer was asked to review the outcome of the tests. She judged the results from both the qualitative and quantitative data to be usable and comparable to similar tests carried out by the UX team.

Table 4. Time consumption per participant for planning, conducting and analysing an AB test.

	Time spent	
Activity	Team 1	Team 2
Planning	2.4 hours	2.4 hours
Conducting test	1.3 hours	1.5 hours
Data Analysis	1.3 hours	1.3 hours
Total	*5 hours*	*5.2 hours*

5.2. Training in AB testing at SenDx

The training followed the same pattern as previously outlined. The four participants were divided into two teams. Each team applied the method on a real-life case. Both cases concerned an internal web tool. One test objective was to determine the wording of a tab and the other was the placement of a search box.

The training materials consisted of a guideline and a template concerning planning and reporting. For more details, see Øvad et al. (2015) and Øvad and Larsen (2016).

Interviews conducted with the developers confirmed the findings from RMED. The training was well received and all developers were surprised by the results: 'I was surprised by the findings – it reopened my eyes on how little I was able to put myself in the end-users shoes and really see things'. They confirmed that the ability to conduct this type of work would give them a larger degree of independence concerning minor UI decisions or initial investigations. SenDX developers had less prior experience in addressing end users than TC and RMED and would benefit from more training in this.

Two weeks later the developers planned, conducted and analysed an AB test. They were once more split into two teams. Each team applied the method on a real-life case from within the company. Test materials were prepared beforehand and the time spent on the tasks can be seen in Table 5.

We observed that developers made extensive use of the support materials. Some minor mistakes were observed. When they were made aware of the problem, they changed their approach accordingly. As previously

Table 5. Time consumption per participant for planning, conducting and analysing AB tests at SenDx.

	Time spent	
Activity	Team 1	Team 2
Planning	1.5 hours	1.7 hours
Pilot test	0.2 hours	0.2 hours
Conducting test	1.6 hours	1.1 hours
Data analysis	0.5 hour Plotted data throughout the test	2.2 hours Incl. typing in data
Total	*3.8 hours*	*5.2 hours*

observed, both teams experienced problems comprehending and analysing qualitative data.

Post-test interviews revealed that they were surprised by the amount of additional data they obtained. When asked if they would perform an AB test again, all developers responded positively. Furthermore, they could see the potential in conducting AB tests:

> When we have these discussions anyway – some requirement discussions or UX discussions – I mean – it takes like an hour anyway to argue about an idea being good or bad … If you are going to do that anyway, then you might as well spend a little bit of extra time and gather some feedback and see where that leads to.

The developers were very much aware of their own performance and were able to evaluate it. One of the teams screened their test participants in order to have as much dispersal as possible and the team was able to support their findings with background information of the test participants.

The findings from these tests were evaluated in two different ways. The software manager evaluated the results from one task and found them acceptable to implement immediately. The person responsible for customer and end-user contact was presented with the results from the other task. These were judged to be valuable and were used as a foundation in further discussions on the design of this workflow.

6. Contextual Interview

The one-day hands-on *in-situ* training approach proved to work well for the AB test training. Next, we returned to the Contextual Inquiry to make the method more available to developers than the original method as described by Holtzblatt, Wendell, and Wood (2005). The developers experienced problems during the analysis phase at TC and found it too time consuming. Pre-training interviews conducted at RMED confirmed this problem. Although the developers were positive towards the method, they did not believe sufficient time would be allocated by the management to conduct the full extent of the method. Hence, the Contextual Inquiry method was restructured and a more streamlined process requiring less in-depth analysis was designed. Detailed guidelines and planning and reporting templates were developed. In addition, a 'cheat sheet' was developed (see Øvad and Larsen 2016). The method is denoted Contextual Interview, to indicate the strong inspiration from Contextual Inquiry (Beyer and Holtzblatt 1997; Holtzblatt, Wendell, and Wood 2005).

The training was carried out at RMED and SenDx. In the following sections, we describe the training and findings at RMED, followed by a short account from SenDx.

6.1. Training Contextual Interview at RMED

The training consisted of a one-day (seven-hour) training session. Five developers divided into two teams participated. They planned, executed and analysed a Contextual Interview on two real-world tasks from RMED's production site. The first team followed the assembly process of a blood gas analyser and the second team followed the testing of an assembled blood gas analyser.

The developers' post-training responses were quite similar to what we observed from the AB test training. The developers were satisfied with the training – especially the hands-on approach and real-world tasks were highly valued.

We still observed problems with in-depth qualitative analysis, but the participants were able to create the physical, sequence and artefact models known from Contextual Inquiry and they succeeded in creating an affinity diagram with one-level headlines. They all felt well equipped for conducting a Contextual Interview again.

Six months after the training, four of the developers planned, conducted and analysed Contextual Interviews on their own.

The Contextual Interviews were conducted as a pre-study for a redesign workshop. They performed four interviews with in-house staff concerning how they handle manual quality control installation on one of RMED's products. The purpose was to support the design phase of this feature in the next-generation products.

The developers split up in two teams, but performed the planning and data analysis together. Furthermore, one of the developers spent some time the day before for planning and inviting participants, and another developer used 30 minutes the day after to finalise the report (see Table 6).

Despite the ability to carry out the Contextual Interview, our observations revealed a lack of understanding of the terminology.

We observed that domain knowledge was a great asset and it enabled the developers to perform a good analysis and consolidate their notes in a proper manner. The analysis furthermore facilitated good discussions.

Table 6. Time consumption per participant for planning, conducting and analysing a Contextual Interview at RMED.

	Time spent (RMED)		
Activity	Team 1	Team 2	Time spent (SenDx)
Planning	1.5 hours		1 hour
Conducting Contextual Interview	1 hour	1 hour	1.5 hours
Data analysis	2.5 hours		4.3 hours
Total	*5 hours*		*6.8 hours*

Post-test interviews revealed benefits similar to that of AB testing. However, the developers did not see the test case for the Contextual Interview suitable for the method. ‘It felt like we bent this [the method] enormously … it had not been the first choice if we were to choose ourselves.’

The interviews also showed inconstancy in the used terminology. This could be due to the large interval between the training and the application of the method (six months).

The Contextual Interview training and Contextual Interviews done by developers were performed at SenDx as well. In this case, the application of the method took place shortly after the training session. Overall, the results corresponded to the results obtained at RMED. However, in this case the tasks for the Contextual Interviews were well-chosen for the method paradigm and we did not observe problems with remembering the terms and the like. Results improved accordingly.

7. Discussion across studies

The one-day, hands-on training, *in-situ* approach has proved applicable. In particular, the real-life tasks were advantageous to apply and clearly motivated the developers. This quickly showed how and where the method could be used in their own work. The structured processes and supporting materials were found helpful as well, together with the opportunity for the developers to ask questions throughout the training sessions.

The time consumption of the methods fit well into the Sprint rhythm (see Table 7). It should be noted that each test was carried out by two or three persons and the total time should, therefore, be multiplied with this number.

We made a triangulation by comparing interview findings with observations and the notes written by the developers to evaluate the training objectives. AB testing and Contextual Interview at RMED and SenDx are compared to the four training objectives below:

Objective 1

> The developer should be able to remember and understand the terminology used when performing the given method.

For AB testing this objective was successfully fulfilled. The developers at both RMED and SenDx were able to actively use the terminology from the method. This was shown both verbally during interviews and observations, and in the filled-in templates.

For Contextual Interview this objective was fulfilled. The developers at SenDx were able to actively use the terminology from the method. This was shown both verbally during interviews and observations, and in the filled-in templates. Problems were found at RMED concerning the description of the mentor/mentee role and the understanding of the different interview styles.

Objective 2

> The developer should be able to judge in which cases the method can be applied. Furthermore, the developer should be able to create a plan for the execution of the method.

This was successfully fulfilled for AB testing. The developers selected two different cases for the AB tests, both matching the AB test paradigm.

This objective was also fulfilled for Contextual Interview. The developers at SenDx selected a case that matched the Contextual Interview paradigm. At RMED all developers displayed knowledge of the method and judged the case they were provided not to match the Contextual Interview paradigm very well. Hence Objective 2 was fulfilled at RMED as well.

Objective 3

> The developer should be able to apply the method to solve a real-life task, together with the ability to analyse the results obtained from this application.

This objective was successfully fulfilled for AB testing. The developers at both sites applied the AB test paradigm to different real-life cases and analysed the results obtained from these tests. However, we observed a lack in their understanding of qualitative data analysis. Yet, the outcome of their tests was approved by external experts.

For Contextual Interview this objective was also fulfilled. The developers applied the Contextual Interview paradigm to different real-life cases independently and analysed the results obtained from the tests. Like above, the qualitative analysis caused problems.

Table 7. The average time consumption per participant when performing the different methods.

	Hours spent			
Tasks	Contextual Inquiry	Focused Workshop	AB testing	Contextual Interview
Planning	3 hours	6 hours	2 hours	1.3 hours
Conducting	4 hours	1.5 hours	1.9 hours	1.3 hours
Analysing	14 hours	5 hours	1.3 hours	3.4 hours
Communicate results	2 hours	2 hours	0	0
Total	*23 hours*	*14.5 hours*	*5.2 hours*	*6 hours*

However, they sufficiently produced the proper models and were able to produce affinity diagrams with one-level headlines.

Objective 4

> The developer should be able to evaluate the results and the usefulness of the obtained data and have the ability to use the results to suggest solutions for further development within the given project.

For AB testing this objective was fulfilled. The developers at both sites were able to evaluate the results and the usefulness of the obtained data. They displayed the ability to use the results to suggest solutions for further development.

For Contextual Interview, this objective was not completely fulfilled. However, the developers were able to evaluate the results and the usability of the obtained data.

8. Conclusion

In this work we have investigated how software developers can be trained successfully to perform certain UX methods. Furthermore, we have applied the constraints of the agile paradigm for software projects, which is predominant in industry. We did this by carrying out extensive experiments in real-life settings in three different companies for three different methods. We modified existing usability and the UX method in an iterative process and by developing supporting materials as well.

We found that it is indeed possible to tailor existing usability and UX methods to fit into an agile, industrial environment.

From initial training sessions in the methods Contextual Inquiry and Focused Workshop, the 'instructor' and 'observe and learn' approach were well received and infused the developers with a high level of confidence. However, our experiences with the 'observe and learn' approach indicated that this approach was too random. Hence, the 'instructor' approach was chosen and used to perform the AB test and Contextual Interview training. We found that using hands-on exercises and real-life tasks extracted directly from the developer teams' sprint backlogs provided motivation, knowledge and confidence in performing the UX work. This approach set the present study apart from most previous ones and we believe it to be a deciding factor for the positive outcome of the study.

Acknowledgements

We wish to thank all participating staff at TC Electronic, SenDx Medical and Radiometer Medical for participating in this work. We thank Søren Vinter Færch at Bankdata and Nis Bornø at Aalborg University for their involvement in the training.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

We thank Aalborg University, Radiometer Medical and the Danish Ministry for Science and Education for funding the research presented here.

References

Beyer, H., and K. Holtzblatt. 1997. *Contextual Design: Defining Customer-centered Systems*. The Netherlands: Elsevier.

Bruun, A. 2010. "Training Software Developers in Usability Engineering: A Literature Review." In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, NordiCHI'10*, 82–91. New York, NY: ACM. doi:10.1145/1868914.1868928.

Bruun, A., and J. Stage. 2014. "Barefoot Usability Evaluations." *Behaviour & Information Technology* 33 (11): 1148–1167.

DIS, I. 2009. *Ergonomics of Human System Interaction-Part 210: Human-centred Design for Interactive Systems*. Geneva: International Standardization Organization (ISO: 9241-210:2010).

Dreyfus, S. E., and H. L. Dreyfus. 1980. *A Five-Stage model of the mental activities involved in directed skill acquisition* (No. ORC-80-2). Berkeley, CA: California Univ Berkeley Operations Research Center.

Eriksson, E., Å. Cajander, and J. Gulliksen. 2009. "Hello World! – Experiencing Usability Methods without Usability Expertise." In *Human-Computer Interaction – INTERACT 2009 SE – 60, Lecture Notes in Computer Science*, edited by T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. Palanque, R. Prates, and M. Winckler, 550–565. Berlin: Springer. doi:10.1007/978-3-642-03658-3_60.

Häkli, A. 2005. *Introducing User-centered Design in a Small-size Software Development Organization*. Helsinki: Helsinki University of Technology.

Holtzblatt, K., J. B. Wendell, and S. Wood. 2005. *Rapid Contextual Design: A How-to Guide to Key Techniques for User Centered Design*. Oxford: Elsevier.

Karat, J., and T. Dayton. 1995a. "Practical Education for Improving Software Usability." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'95*, 162–169. New York, NY: ACM Press/Addison-Wesley. doi:10.1145/223904.223925.

Karat, J., and T. Dayton. 1995b. "Practical Education for Improving Software Usability." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 162–169. New York, NY: ACM Press/Addison-Wesley.

Kjeldskov, J., M. B. Skov, and J. Stage. 2004. "Instant Data Analysis: Conducting Usability Evaluations in a Day." In *Proceedings of the NordiCHI'04*, 233–240. Tampere: ACM.

Krathwohl, D. R. 2002. "A Revision of Bloom's Taxonomy: An Overview." *Theory Into Practice* 41 (4): 212–218.

Krueger, R. A., and M. A. Casey. 2002. “Designing and Conducting Focus Group Interviews.” *Social Analysis, Selected Tools and Techniques*, 4–23. http://www.eiu.edu/ihec/Krueger-FocusGroupInterviews.pdf.

Latzina, M., and B. Rummel. 2003. “Soft(ware) Skills in Context: Corporate Usability Training Aiming at Cross-Disciplinary Collaboration.” In *Proceedings 16th Conference on Software Engineering Education and Training, 2003 (CSEE&T 2003)*, 52–57. Los Alamitos, CA: IEEE.

Lewis, C. H. 1982. *Using the “Thinking Aloud” Method in Cognitive Interface Design*. Technical Report IBM RC-9265. Yorktown Heights, NY: IBM Watson Research Center.

Mathiassen, L. 2002. “Collaborative Practice Research.” *Information Technology & People* 15: 321–345.

Medlock, M. C., D. Wixon, M. Terrano, R. L. Romero, and B. Fulton. 2002. “Using the RITE Method to Improve Products: A Definition and a Case Study.” Presented at the Usability Professionals Association, Orlando, FL.

Nielsen, J., R. M. Bush, T. Dayton, N. E. Mond, M. J. Muller, and R. W. Root. 1992. “Teaching Experienced Developers to Design Graphical User Interfaces.” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 557–564. New York, NY: ACM.

Nielsen, J., and R. Molich. 1990. “Heuristic Evaluation of User Interfaces.” In *CHI ‘90 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 249–256. New York, NY: ACM.

Øvad, T., N. Bornoe, L. B. Larsen, and J. Stage. 2015. “Teaching Software Developers to Perform UX Tasks.” In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, 397–406. New York, NY: ACM.

Øvad, T., and L. B. Larsen. 2014. “Experiences from Training Agile Software Developers in Focused Workshops.” In *Proceedings from the International Conference on Interfaces and Human Computer Interaction*, 355–359. IADIS Press.

Øvad, T., and L. B. Larsen. 2015. “The Prevalence of UX Design in Agile Development Processes in Industry.” In *Proceedings of the 2015 Agile Conference (Agile 2015). Presented at the Agile2015*, 40–49. Washington: IEEE Computer Society Press.

Øvad, T., and L. B. Larsen. 2016. “Templates – A Key to Success when Training Developers to Perform UX Tasks.” In *Integrating User Centered Design in Agile Development*, edited by G. Cockton, M. Lárusdóttir, P. Gregory, and Å. Cajander. London: Springer.

Patton, M. Q. 1990. *Qualitative Evaluation and Research Methods*. Beverly Hills, CA: Sage.

Polson, P. G., C. Lewis, J. Rieman, and C. Wharton. 1992. “Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces.” *International Journal of Man-Machine Studies* 36: 741–773.

Radiometer Medical ApS. 2015. http://www.radiometer.com.

Rubin, J., and D. Chisnell. 2008. *Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests*. Indianapolis, IN: John Wiley & Sons.

SenDx. 2015. https://www.linkedin.com/company/1217829.

Sutherland, J., and K. Schwaber. 2011. The Scrum Guide. Defin. Guide Scrum Rules Game.

TC Electronic. 2015. http://www.tcelectronic.com/.

Yin, R. K. 2010. *Qualitative Research from Start to Finish*. New York: Guilford Press.