Customer and end-user participation in agile software development

Master thesis

Jakob Nymann Holgersen
Mathias Huse Jensen
Pelle Ulmer Jørgensen

Aalborg University
Department of Computer Science
Selma Lagerlöfs Vej 300
DK-9220 Aalborg Ø

Software
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Customer and end-user participation in agile software development

Theme:
Human-Computer Interaction
Master thesis

Project Period:
Spring 2019

Project Group:
hci1025f19

Participant(s):
Jakob Nymann Holgersen
Mathias Huse Jensen
Pelle Ulmer Jørgensen

Supervisor:
Jan Stage

Copies: 3

Page Numbers: 69

Date of Completion:
07-06-2019

Abstract:

This master thesis concerns customer and end-user participation in agile software development. It is based on observed similarities and differences between development and evaluation in research and in practice. To enable this investigation, we conducted two studies into including the end-user in a research setting and a study on how customer and end-user participation is done in practices. This is described in three distinct academic papers. The first paper presents a case study where Scrum was combined with a participatory design method. The second paper investigates two different remote asynchronous usability testing methods, these being the use of diary and continuous updates in addition to UCI reports. The third paper investigates Danish software companies regarding the practical application of customer and end-user participation in modern development methods. This master thesis shows that an end-user takes on different roles during development and evaluation. These being consultative, informative and participative in development against consultative and participative in evaluation. The preferred techniques used by the companies included: Interviews, Demonstrations, User stories and Qualitative usability testing. While obstacles included: Lack of time, Geographical challenges and Lack of staff. In relation to techniques used and obstacles experienced in the research setting, these where similar to those identified in practice.

Summary

This master thesis contains detailed research into how customer and end-user participation can be facilitated both in research and practice. This master thesis is a summary report of three academic papers each detailing a research question, contributions and conclusions related to the specified research question. Furthermore, the report contains a description of the overall research question that each paper is contributing to, together with overall conclusions draw based upon the contributions from each paper.

Contribution 1 found in section A.1 in Appendix A, details the theory and usages of an adapted development method. This method is a combination of Scrum and the participatory design method MUST. This method utilizes the power of an agile and iterative nature, with the strengths of techniques and methods supporting end-user participation specified from the theory of MUST. This paper shows, through a case study revolving around diabetics and patient empowerment, the feasibility of combining Scrum and MUST in order to facilitate end-user participation throughout the entire development process, except implementation activities.

Contribution 2 found in section A.2 in Appendix A, details research into how two remote asynchronous usability methods can be employed. The two methods investigated are: inclusion of a diary and providing continuous updates. This paper shows, that having a diary, contributes to finding usability problems in addition to also provide knowledge about how the users used the application. Furthermore, the groups who did not receive updates continuously reported the same usability problems throughout the testing period, where the groups receiving updates did not.

Contribution 3 found in section A.3 in Appendix A, details a survey conducted with companies from the Danish software industry. This survey's purpose was to map and understand the techniques and methods use to facilitate customer and end-users participation. This includes; company demographics, at what stage each technique was used, when participation was deemed most useful, the level of current customer and end-user participation and the obstacles they face. This paper shows, that Scrum was a dominating development method since 79.2% of respondents used it. Furthermore, 51.5% of respondents thought that there were insufficient end-user participation in their current development process. The most used

techniques for participation were Interviews, Demonstrations, User stories and Qualitative usability testing. The primary identified obstacles for additional customer and end-user inclusion were Lack of time, Geographical challenges and Lack of staff.

This master thesis shows that the end-users had different roles depending on if they were involved in development or evaluation. When the end-users were included in development they had a participative, consultative and informative role as they were involved in meetings during each sprint. Where when the end-users were evaluating the application they had a participative and consultative role. As they performed the remote usability testing, where they reported usability problems and provided feedback though the UCI reports and diaries. Furthermore, when looking at the techniques used and obstacles identified in practice, these where similar to those experienced in the research setting. Among the most prominent techniques used for facilitating customer and end-user participation in practice were Interview, Demonstrations, User stories and Qualitative usability testing. This is similar to the Interviews, Walkthroughs and Qualitative usability testing performed in the research setting. In terms of obstacles, Geographical challenges, Lack of Time and customer and end-user participation Does not fit development process were all challenges that were accommodated for in the research setting.

Preface

This master thesis has been made by hci1025f19, 10th semester Software students from Aalborg University. The focus of the master thesis is to investigate how customer and end-user participation can be facilitated, during development and evaluation, both in a research setting and how this is done in practice.

This master thesis is a summary report of three academic papers each detailing a research question, contributions and conclusions related to the specified research question. Some of the detailed work was carried out on our 9th semester and is documented in [19].

The master thesis has been developed in cooperation with the project supervisor, Jan Stage, who we express gratitude and thanks towards.

Aalborg University, 7th June 2019.

Pelle Ulmer Jørgensen
pjarge14@student.aau.dk

Jakob Nymann Holgersen
jholge14@student.aau.dk

Mathias Huse Jensen
matjen13@student.aau.dk

Contents

1 Introduction

In 2001 twelve principles and four values constituting the cornerstones for what agile development is, were described in The Agile Manifesto [8]. One of the values from The Agile Manifesto, is "Customer collaboration over contract negotiation", mean that the role of a customer needs to be taken further than only specifying requirements and be a more active role in the development process. The reasoning for promoting customer collaboration over contract negotiation is the need for a higher level of partnership in the development process, rather than sacrificing efficiency by spending a large amount of time negotiating a contract. This contract can be exploited as a shield for the customers or as a weapon for the developers, such that the features developed are the ones exactly specified by the contract, and not necessarily what they were looking for. However, a better relationship between the customers and the developers are a 2-way partnership, where both parties work together to achieve the best product. Besides this, The Agile Manifesto does not specify any methods or techniques to facilitate customer and end-user participation during the development [8].

Some of the development methods which have emerged and adopted the values of The Agile Manifesto, are Scrum and Extreme Programming (XP). In XP an On-site customer is the representative from the customers, where in Scrum this role is fulfilled through a Product Owner. However, these representatives have multiple responsibility, for an example in Scrum, the Product Owner have the sole responsibility for managing the Product Backlog and being responsible for providing guidance and support towards the project's success. Hence, being a Product Owner is usually a full-time job [32][36]. These responsibilities and being a full-time job, means that the Product Owner is not necessarily able to take the end-users' needs into account. This matches studies that show that customer representatives, such as a Product Owner, only have a limited understanding of the users' needs, and that they might not be the actual users of the system [27][33][34]. In addition to this, Scrum has no clear techniques for facilitating end-user participation [8] which forces companies to use other methods if they want to include the end-user.

Inadequate user participation can lead to adverse consequences such as: pressure to overcommit by the developers, ensuring feedback from the user, the loss of productivity as a consequence of not knowing the specifics of what needs to be developed and problems in gathering, clarifying and prioritizing requirements [18]. In addition to this, studies have

shown a positive effect of user participation through development, as enhancing user participation could lead to improved quality design decisions and enhance the user's commitment to, and acceptance of, the application [6][14]. Participatory design is a different approach to user participation, where there is active end-user participation throughout the development and not just a person representing the end-users, such as a Product Owner or On-site customer. The goal of participatory design is to involve end-users, stakeholders, designers and researchers in the design process, such that the product is created to satisfy the needs of its user-base [4].

Beside ensuring customer and end-user participation during development, evaluation with the end-user is considered essential for a products success. Especially usability evaluations has shown to have a great effect [7][11][20]. There are different ways to perform usability evaluation. This includes traditional laboratory evaluation, field evaluation and remote usability evaluation. As software development are becoming more globalized, developers, users and usability evaluators can be distributed across continents and time zones. These changes in software development imply that new methods and approaches are needed to facilitate the evaluations, as tradition laboratory- and field based evaluation does not take time and distance into account [1][29]. Remote usability testing are able to account for both time-zone differences and distance, with the added benefit of the users ability to perform the usability test in a real use context. To account for time, the method can be performed as asynchronous, meaning the usability evaluation can be performed at a different time and distance relative to the evaluators, or synchronous, meaning that the evaluators and testers are separated only in space.

With Agile development methods dominating how software is developed, it would be interesting to investigate, how customers and end-users can be included in development and evaluation, both in research and in practice.

1.1 Research questions

In this master thesis we differentiate between customers and end-users, where a customer is defined as the company / person receiving the developed service / software while the end-user is defined as the user, who ultimately uses the product. This leads to the following overall research question:

> **Overall research question: How can customer and end-user participation be facilitated in a Scrum process during development and evaluation in a research setting and in practice?**

We want to examine if combining an agile development method with a participatory design method could ensure facilitation of end-user participation by utilizing techniques from both

methods. This leads to the first research question:

> **1st Research Question: How can Scrum be combined with a participatory design method to facilitate end-user participation in every step of the development, except implementation activities?**

This research question addresses specifically how to facilitate end-users' participation during development by combining an agile development method and a participatory design method. This was done by a case study, in which we developed an android application using the adapted development method. The case is described in section 1.2 and a description of the application can be seen in Appendix B. To get a better understanding of how to utilize the end-user when evaluating the application, we chose to investigate how users can participate in a longitudinal usability evaluation of a product. This leads to the second research question:

> **2nd Research Question: How can users participate in a longitudinal evaluation of the usability of a product from an agile development process**

Given the nature of the case, described in section 1.2, it was essential that the evaluation was performed in a real life context. In addition to this, to evaluate certain features and functionalities the evaluation would have to be performed over a period of time, in order to reflect how the application would be used in a real life context. This meant that performing a longitudinal evaluation would be favorable. Furthermore, performing longitudinal testing could also eliminate the issues of usability problems related to first time usage, since the longitudinal test are able to reveal change and growth in a users behavior [22].

This research question investigates different methods as to how to perform a longitudinal usability evaluation with end-users. While the two first research questions takes place in an research environment, we want to investigate customer and end-user participation in practice. Hence, we wanted to investigate how, when and which methods that are used to facilitate customer and end-user participation during software development with companies from the Danish software industry. This leads to the third research question:

> **3rd Research Question: How is customer and end-user participation facilitated during agile development in the Danish software industry?**

1.2 The case study

This case study is used as a basis for the first two research questions. The case study revolves around diabetes type 1 and type 2, more specifically the daily life of a diabetic. The reasoning for conducting a case study concerning diabetics, is that the development of the

application had a focus on end-user participation and patient empowerment. End-user participation can help improve the quality of the application, but also empower the end-user to give opinions and ideas to what design and which features are important. This is in-line with the process of empowering patients, which in term can help improve their own treatment. Furthermore, the subject of diabetes is of personal interest to the authors. In addition to this we were able to acquire contact with a health professional. Diabetics are seen as the primary stakeholders for the case study. Furthermore, one of the diabetics is chosen as the Product Owner for the case study. Diabetes is a disorder where each individual has to handle it differently, which means that the challenges faced as a diabetic are different for everyone. Examples of these challenges are aspects of daily life such as: Diet and when and how much medicine to take.

The case study focuses on developing an application to help diabetics acquire knowledge and help manage their daily life with diabetes. The application needs to provide a better understanding of diabetes and assist the users in making informed choices that can improve their life with diabetes. This will contribute to the users becoming more empowered and in control of their own treatment, i.e. support certain patient empowerment factors [5][25]. A description of the application can be seen in Appendix B.

2 | Contributions

In this chapter we present the contribution of each of the three academic papers, which can be found in Appendix A. Table 2.1 shows what areas of research each paper and the research question we wanted to study.

	Development	**Evaluation**
Research	Paper 1 - RQ1	Paper 2 - RQ2
Practice	Paper 3 - RQ3	

Table 2.1: Papers' area of research

2.1 Contribution 1

Jakob Nymann Holgersen, Mathias Huse Jensen, Pelle Ulmer Jørgensen. Combining Scrum with participatory design: A case study of end-user participation. *Department of Computer Science, Aalborg University*. Aalborg 2019.

This contribution presents a case study, with an adapted development method, where Scrum was combined with a participatory design method, MUST, to further ensure end-user participation in every step of development, except implementation activities. Only the end-user were included in this study. The reason for this, was that in this case study the primary stakeholders were diabetics, thus per our definition of customer and end-user, there are no customer stakeholder. The study documents the developers experiences with end-user participation during development of a mobile application for diabetics, with a focus on patient empowerment. Patient empowerment refers to the philosophy of giving the ability to act and take informed choices regarding health treatments back to the patient [25]. The feasibility of combining Scrum and MUST is based on whether it allows and facilitates active end-user participation throughout the entire development, except implementation activities.

We conducted a case study, as described in section 1.2, using the adapted development method. This development method was iterative and divided into three sprints inherited from Scrum. Each sprint contained activities from both Scrum and MUST. This included all events and artifacts from Scrum and the four phases from MUST with selected techniques. The focus of the case study was to ensure a high level of end-user participation in each of the sprints. This meant that the techniques which promoted feedback and active participation were weighted higher than non-promoting. Hence, Walkthroughs and Interviews, were used to involve end-users and gain information that later were used to produce a Product Backlog in each sprint. A Product Owner was involved through both the Walkthroughs and Interviews, but also in the Sprint Review meetings, where the Product Backlog containing features and functionalities requested by the end-users was reviewed. Before development began, a phase not specified by either Scrum or MUST was included. This phase, named the pre-sprint, was used to identify and analyzed the problem domain that the case revolves around. This was done in order to gain knowledge about the different concepts of the case. The pre-sprint contained techniques from the Initiation phase from MUST, which focuses on planning and mapping the project with techniques such as Baseline Planning and Document analysis. The product of the Initiation phase was a Project charter, which outlines the project and the techniques that were going to be used in order to successfully achieved the desired participation of end-users. The experiences of deploying the adapted development method have been documented using daily diary entries written by the development team.

Based on the case study we concluded that we experienced a high level of end-user participation throughout the entire development process, with the exception of the implementation activities. It was shown that the techniques used were able to successfully facilitate end-user participation. Through development we learned that some of the activities chosen were more important than others, when these were part of an iteration. For example, we learned that iterating a SWOT analysis every sprint were unnecessary, where iterating Walkthrough and follow-up Interview were critical for the development. Furthermore, we experienced that by having end-user participation, we were able to continuously validate the implemented utility and further generate new requirements for the system.

2.2 Contribution 2

Jakob Nymann Holgersen, Mathias Huse Jensen, Pelle Ulmer Jørgensen. A longitudinal evaluation of a mobile application using two remote asynchronous usability test methods. *Department of Computer Science, Aalborg University*. Aalborg 2019.

This contribution presents a longitudinal evaluation of the mobile application, developed through the first paper, using two remote asynchronous usability test methods. One method is testing whether reporting usage and usability problems, through a diary, have an effect on the User-reported Critical Incidents (UCI) reports the users create and the quality of the feedback. The other method is whether a continuously updated application during testing,

have an effect on the UCI reports created by the users. Additionally this paper investigates whether the test participants are able to rate the severity of usability problems themselves.

The 16 users participating in the evaluation were all defined as end-users of the system. The reasoning was that the primary users of the system were diabetics, thus per our definition of customer and end-user, there are no customer stakeholder. The 16 users were divided into four different conditions: Diary and updates, only updates, only diary and no diary no updates. At the beginning of the test period the users received an instruction guide, detailing what a usability problem is by a description and examples. In addition to this, a step-by-step instruction on how to report the usability problems in the application were given. To help the users get familiar and use the application, all users received daily assignments, both as SMS and email, throughout the two-week test period. In addition to this, push-notifications were implemented and set to remind the users three times a day, morning, mid day and evening, to use the application and report encountered usability problems. Lastly follow-up interviews were conducted with the users to investigate and clarify interesting observations.

During the evaluation, challenges performing remote asynchronous usability testing were discovered. First, the users were not reading or understanding the instruction guide: At the start of the test period each user received extensive instructions explaining what a usability problem is and how they could be identified. Despite this information, results indicate that the users did not read or understand these instructions. Second, lack of commitment: Given the nature of remote usability testing, it became apparent that the users commitment to reporting and completing the daily task were lacking. Third, ending the test period early: Some users told, in the follow-up interviews, that they had ended the test period early without contacting us. This have had an effect on the remote usability testing, since we were not aware that we had to find users to substitute the users that ended testing early. Consequently, these challenges influenced the amount of UCI and diary entries received.

Based on the results from the usability evaluation, we showed that having a diary, contributed to identifying usability problems, in addition to providing knowledge about the usage of the application. Furthermore, we showed that groups without updates reported the same usability problem multiple times throughout the test period, where the groups who received updates which fixed the issues did not. In addition to this, it was shown that the users, were able to rate 59% of the identified usability problems similar to the rating given by evaluators. This result was surprisingly low, in comparison to related work and cannot be related to a specific variable or potential outliers.

2.3 Contribution 3

Jakob Nymann Holgersen, Mathias Huse Jensen, Pelle Ulmer Jørgensen. Customer and end-user participation in software development: A survey of the Danish software industry. *Department of Computer Science, Aalborg University*. Aalborg 2019.

This contribution presents a survey of the Danish software industry with a focus on how, when and which methods are used to facilitate customer and end-user participation in development. This is done as an exploratory study to gain a picture of how Danish software companies are handling customer and end-user participation and by that, how they alter their development method to ensure this. A customer is defined as the company / person receiving the developed service / software while the end-user is defined as the user, who ultimately uses the product.

We conducted five semi-structured interviews with developers from three different software companies to gain knowledge of what questions the survey should contain. We then created a pilot questionnaire which was sent to the five interviewees. Through this pilot test, we received two responses containing revisions. Based on the revisions the final version of the questionnaire was created. 102 companies were contacted, where 67 of these agreed to participate and received the questionnaire. The size of the companies ranged from 12 to 4.100 employees. Beside the companies, the questionnaire was shared on Social media and relevant forums. All of the recipients were asked to forward the survey to any interested or relevant parties, reaching upward of 67.000 people. In total 53 responses were received, which were analyzed.

Based on the results from the survey it was discovered that Scrum was a dominating development method, since 79.2% of respondents used it. Looking at which development methods that are being used worldwide, CollabNet VersionOne [10] publish an annual report describing the state of agile development. Throughout 12 reports spanning from 2006-2017, Scrum is the most prominent method. Scrum was used by 56% in 2017, with an increased by 16% from 2006. Similar in our results, Scrum is the most prominent agile development method with 79.2%. In relation to the report from CollabNet VersionOne, the difference in the number of people using Scrum can be explained by the respondents only being able to answer one method in the survey from CollabNet VersionOne, while being able to answer multiple development methods in our survey.

It was discovered that for the surveyed companies 28.2% said that there was insufficient customer participation and 51.5% said that there was insufficient end-user participation. The perceived lack of customer and end-user participation did not have a strong correlation to either the product type produced by the company, the current job position of the respondent or the development method used by the company. The respondents were asked to select the phases of development they found customer and end-user participation to be most useful in. An interesting observation here was that the difference in responses for the four highest voted phases (Concept, Analysis, Design and Test) were at most 12.8% for customers and 12.1% for end-users. Consequently customer and end-user participation is equally distributed across these four phases and therefor there is only a slight difference in perceived usefulness of participation in each phase, with the exception of the Implementation- and Maintenance phase. In addition to this, it was shown that the most used techniques for customer and end-user participation were Interviews, Demonstrations, User stories and Qualitative usability testing. An interesting observation of the use of Demonstrations was the

difference in terms of including the customers and end-users. Here the customers were used by 30 (76.9%) respondents compared to 16 (48.5%) respondents with end-users. Similar to this was the usage of User stories where 24 (61.5%) respondents used User stories for customers and 10 (30%) respondents used it for end-users. By this it can be seen that companies include their customer more than their end-users.

It was discovered that the primary identified obstacles for additional customer and end-user participation were Lack of Time, Geographical challenges and Lack of staff. The respondents that chose Lack of Time as an obstacle had chosen Scrum and Kanban as their development methods, with a few also choosing Feature-driven development. Furthermore, 8 out of 10 respondents that chose Lack of time as an obstacle for customer participation worked for a company with 50 or more employees. 5 out of 5 that chose Lack of staff as an obstacle for end-user participation worked for a company with 50 or more employees. Another interesting result is the difference regarding the obstacle: Customer and End-user participation Does not fit development process. This obstacle was reported by 10.3% when working with the customer while 21.2% reported this obstacle regarding the end-user. 6 of the 7 respondents that answered Does not fit development process regarding the end-user worked with Scrum. All respondents that chose this obstacle for the customer also worked with Scrum. Scrum uses the Product Owner to include the customer, while having no specific method for including the end-user [35][8].

3 | Research Methods

In this chapter we will describe and discuss the research method use in the difference papers. This will include the strength and weakness of the given method and how we attempted to minimize the weaknesses. The research methods used are in the same order as the papers' research question and contributions have been presented.

3.1 Case study research

A case study is an empirical inquiry that investigates a contemporary phenomenon in-depth and within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident [38]. Hence, when we want to understand a real-life phenomenon in-depth, we have to understand the important surrounding contextual conditions, which case studies can be used for [16][31][38]. This close examination can then be used to build understandings, generate theories and hypothesis, present evidence for the existence of certain behavior or to provide insight that would otherwise be difficult to gather [23][31]. In Table 3.1 the benefits and challenges of case study research is shown.

Benefits	**Challenges**
• Help describe or explain behavior • Able to investigate phenomenon in depth within its real-life context • In-depth examinations of individual cases can help generate and build understanding of new theories and hypotheses	• Case studies are time consuming • Making broad generalizations based on a study of one case should be made with caution • Identifying appropriate participants may be more important than for larger studies

Table 3.1: Benefits and challenges of case study research [23][38]

Robert K. Yin [38] and Lazar et al. [23] identifies four different types of case studies: *Exploration* is when the goal is to understand novel problems or situations, often with the hopes of informing new designs. *Explanation* is when the goal is to develop models that can be used to understand a context of technology use. *Description* is when the goal is to document a system, a context of technology use, or the process that led to a proposed design.

Demonstration is when the goal is to show how a new tool was successfully used.

In Contribution 1, which can be seen in Appendix A.1, we have used the exploration type, since our goal was to understand the life of a diabetic. To account for the challenge of identifying the appropriate participants, we got in contact with a equal amount of type 1 and type 2 diabetics. In addition to an equal distribute from each type, it was important to represent different stages of the diabetics' life. This was done by ensuring an equal distribution in the participants' demographics in terms of gender, age, treatment and time since diagnosed. This created a base for obtaining a broad knowledge about the problems and situations these diabetics were facing. To compensate for the challenge of case studies being time consuming, the contact with the primary stakeholders were limited to one week of each three-week sprints, since limiting the contact was necessary to be able to ensure short sprints. This helped us in structuring the contact, such that this activity did not blow out of proportion. However, by limiting the time for end-user participation we were limited in the type and number of activities we could perform, which forced us to choose between techniques that could have been beneficial.

3.2 Interviews

Interviews is a technique for direct conversations with fewer participants, which then can provide perspectives and useful insight that surveys might miss [21][23]. In Table 3.2 the benefits and challenges of interviews are shown.

Benefits	**Challenges**
• Gather data that would otherwise be very hard to capture • Deep knowledge about a specific domain • Able to explore interesting answers you might not expect on the fly	• The choice of interview type has an impact on how the participants can answer • What you ask and how you ask it has a great impact • Deciding what is important and what is not when analyzing • Managing potentially unbounded discussions • Identifying who to interview can be a challenge

Table 3.2: Benefits and challenges of interview [23]

Interviews are divided into three different types [23][21]: *Fully structured* interviews uses a strict script to ask questions in a specific order [23]. *Semi-structured* interviews uses a script to ask questions, but in comparison to fully structured, interviewers may go off-script and ask follow-up questions if the interviewee mentions something of interest [23]. The *free-form unstructured* interview does not use a script, but starts of with an initiating questions or topics, from which the interviewee takes over. This form of interview allows the interviewee to talk about everything they want, with the interviewer only guiding the interviewee back on the initial topics [23].

Through our research, interviews were used in all three contributions, which can be seen in Appendix A. We used the semi-structured type, as our goal was to gain an insight in the domain of the interviewee, hence it was important to be able to go off script and ask follow up questions if the interviewee said something interesting. The choice of semi-structured interview had a positive effect on the challenge, Interview type having an impact on how the participants can answer, as the interviewer were able to ask the interviewee to elaborate and change questions while performing the interview. An example from contribution 1, is when interviewing diabetics, the goal was to get an understanding of their daily life, and as such asking the interviewee to elaborate on interesting subjects was especially effective. To account for the challenge of identifying who to interview, it was important to equally distribute our interviewees based on demographics. An example from contribution 3, is the interviews that were used as a basis for our questionnaire. We interviewed five different developers from three different companies, where the companies differ in size and type of product they produces.

3.3 Diary

A diary is a document made to record events that occur during a specific event, for example documenting the development of a system [23]. A diary can contain a range of different types of entries. It can range from simple entries describing activities to personal reflections. Diaries are useful for documenting measurements that are unable to be collected by observations or experiments and usage patterns that can span multiple systems, location and environments [23]. A diary can be used as an non-intrusive way of collecting data, as the user on their own can choose when to record data [9][23][26][30].

Table 3.3 describes the benefits and challenges of using a diary.

Benefits	Challenges
• Good for collecting user-defined data (e.g., when a user intended to perform an action but did not do so)	• Time recording may be less accurate than in a controlled laboratory setting or automated data collection
• Good for understanding how individuals utilize technology in non-workplace, non-controlled, or on-the-go settings	• Participants are sometimes not introspective and not aware of the specifics of what they are doing; they may therefore have trouble recording it in a diary entry
• Good for understanding the "why" of user interaction with a technology or any technology phenomenon	• Participants may not follow through and record a sufficient number of entries
• More accurate time recording than in a survey	• Since data is both qualitative and quantitative, data analysis may take a long time
• Good for collecting data that is fluid, and changes over time (such as time, mood, perception or response)	• Hard to strike a balance between a frequent-enough series of diary entries and infringement on daily activities (user participation may then trail off)

Table 3.3: Benefits and challenges of a diary [23]

In contribution 1, Appendix A.1, a diary was used to document the usage of an adapted development method used during the development of a mobile application. The diary in this contribution was written by the developers during the last 30 minutes each day. By doing this it was ensured that there would be sufficient diary entries and that creating these entries would not infringe on regular activities. Additionally, a template for a diary entry was created such that it was ensured that the data collected would be introspective. Each diary entry was analyzed at a later time to find interesting observations regarding the development process.

In contribution 2, Appendix A.2, a diary was used in one condition for remote asynchronous usability testing. In this contribution a diary was given to users performing the remote asynchronous usability testing. The purpose of the diary was an additional method to collect usage data about the application and provide users with a method of communicating possible problems, suggestions and feedback. The diary was following a template to ensure the users would be able to provide insight into their usage and document good and bad experiences. To ensure a sufficient number of entries and not interfering with daily activities, a single reminder at 8 pm reminding the users to fill out the diary was given.

3.4 Survey

A survey consists of a set of well-defined questions which a respondent gives answers to [23][37]. Surveys are typically self-administered meaning that no researchers are present when a respondent answers a survey. A survey allows for statistically accurate estimation of a population, given that the respondents of the survey is part of the target population the

survey is investigating. Surveys can be used to measure a variety of different aspects such as awareness, intent, feedback, user experience and more [23][28][37].

Benefits	Challenges
• Collect large amount of answers • Gain answers from large geographical areas • Cost of a survey is relatively low • Easy to distribute fast online	• Data received are shallow data • Data can be biased • Not possible to ask follow up questions • Delivering the survey online can end in spam filters

Table 3.4: Benefits and challenges of survey[3][12][13][17]

In contribution 3, which can be seen in Appendix A.3, a survey was conducted regarding customer and end-user participation in companies from the Danish software industry. The survey was conducted over a period of 12 weeks and distributed over email. To try and alleviate the challenges of the survey ending in spam filters, relevant participants were contacted directly over the phone, asking them to participate. By doing this we ensure that the participants are aware of the survey and by that are able to find it in their inbox. In this survey to accommodate some of the other downsides of a survey, we made responses anonymous, such that the respondents would answer honestly, as there would not be a direct connection between them and their answers.

3.5 Remote asynchronous usability testing

Lazar et al. [23] denotes usability testing as a research method. Usability testing denotes testing a users ability to navigate and use the functionalities of a system. Usually representative users are attempting to perform representative tasks in representative environments on the system. The system could be an early prototype or a more finished version of the system [23][24]. Remote usability testing is to perform usability testing but separating the users and evaluators by time, space or both [2][23]. Separating by only space but still having users and evaluators interact is remote synchronous usability testing, and separating by both time and space is remote asynchronous usability testing [15][23]. The system the user is testing need to be monitored so the evaluators are able to know what the users are doing. Monitoring can be performed in different ways such as video, audio or transmitting other contextual usage data via an internet connection [15][23]. Table 3.5 shows the benefits and challenges of remote usability testing.

Benefits	Challenges
• Access to a greater number of participants • Possible to conduct multiple usability tests at the same time • Less time-consuming than traditional usability testing • Participants have more flexibility when participating since it is easier to accommodate participants schedule • Easy collection and analysis of click stream data	• Difficult or impossible to track nonverbal cues • Hard or impossible to provide instructions when problems occur • Not possible to ask probing questions based on what occurs • Possible to miss the context of what happen during testing

Table 3.5: Benefits and challenges of remote usability testing [23]

In contribution 2, in Appendix A.2, different methods of remote asynchronous usability testing were studied. The reason for this was to investigate if adding additions to remote usability testing would improve the data received from this. Furthermore, by providing additions to the remote asynchronous usability testing it might be possible to alleviate some of the challenges. One method was to include a diary in which the users described what they had used the system for throughout the day and what had happened when using the system. The other method was to continuously update the system while the users were testing it to investigate if removing some of the identified usability problems, would influence the users testing the system. The study was performed as a longitudinal study spanning a period of two weeks. The idea of providing a diary was that the diary would provide an additional way of providing feedback. Furthermore, the feedback in the diary could be longer than the UCI report feedback and contain more explanations of a user's experience. This would help with the challenge of missing the context of problems as the users would be able to express them-self in detail through the diary. Additionally, by providing the diary, we would better be able to form a basis for additional probing question regarding the users experience with the system.

4 | Conclusion

In this chapter we conclude upon the overall research question. This is done by concluding on each of the three research questions based on their contributions, which can be seen in Appendix A. Hereafter, limitations of the research is presented. Lastly, future works is discussed.

4.1 Research Question

> **1st Research Question: How can Scrum be combined with a participatory design method to facilitate end-user participation in every step of the development, except implementation activities?**

The adapted development method is based around Scrum sprints, where the phases of MUST is integrated. The adapted development method starts with the MUST initiation phase where the project is planned. Following this, the Scrum sprints start and during the first week of each sprint the In-line- and In-depth analysis phase of MUST is performed. The remaining two weeks of each sprint consists of the Innovation phase of MUST alongside Scrum activities and techniques.

Based on the results of this case study, we can conclude that the adapted development method did facilitate a high level of end-user participation in the development process. It was shown that the techniques and activities used in the other phases, especially Walkthrough and follow-up Interviews, facilitated end-user participation. As this provided us iteratively with a deep level of knowledge, of the context in which the application should be used and what functionality the application should have. Furthermore, the participation of end-users enabled new requirements for the system to be generated continuously, together with validation of the existing utility throughout the entire development process.

It was shown that iterating the In-line analysis phase, each sprint, was redundant. As the techniques in the In-line analysis phase helps describe the environment the application is

part of and only minimal changes occur in the environment in a three week time period. Eliminating the In-line analysis phase in each sprint, would allow more time for deeper analysis in the In-depth analysis phase, this includes time for additional techniques.

2nd Research Question: How can users participate in a longitudinal evaluation of the usability of a product from an agile development process?

The users for the longitudinal evaluation were recruited through the local diabetes association's news email and diabetes associated Facebook groups. The longitudinal evaluation was performed by grouping the users into four different groups. Each group would correspond to a specific condition, these being whether the application had a diary feedback functionality and if the application would receive continuous updates. All conditions had access to a page in the application created specifically for creating UCI reports. Each user was given an instruction guide explaining what a usability problem is, how to report it and how to install the application. The users received daily tasks they could perform, in addition to their normal use of the application.

Based on the result from the longitudinal evaluation, we learned that having a diary as an additional method of providing feedback, did contribute to finding usability problems in addition to provide knowledge about the usage of the tested application. Furthermore, the longitudinal evaluation showed that groups that did not get updates during the longitudinal evaluation, reported the same problems throughout the entire test period.

In addition to this challenges with longitudinal remote asynchronous usability testing were discovered. These challenges were that the users had a lack of commitment to testing. That the users did not read or had problems understanding the material explaining what a usability problem is and how they should report it. Furthermore, some users did not inform when they decided to end testing before the intended end date. These challenges might be related to insufficient screening of the users before the test period started.

3rd Research Question: How is customer and end-user participation facilitated during agile development in the Danish software industry?

Based on the results from the survey, it was discovered that Scrum was a dominating development method, since 79.2% of respondents used it. The results show that 51.5% of respondents thought that there was insufficient end-user participation in their current development process and 28.2% thought that there was insufficient customer participation. The most used techniques for customer and end-user participation were Interviews, Demonstrations, User stories and Qualitative usability testing. However, a Chi-square test revealed no significant difference, meaning that the techniques used to facilitate customer and end-user participation had no significant correlation to the estimated level of customer and end-user participation. The primary identified obstacles for additional customer and end-user participation were Lack of Time, Geographical challenges and Lack of staff.

Overall research question: How can customer and end-user participation be facilitated in a Scrum process during development and evaluation in a research setting and in practice?

To conclude upon the overall research question, we conducted three studies on customer and end-user participation. We compare the results from each of the three papers. Table 4.1 shows an overview of how each paper contributes to their area of research. We compare Paper 1 and Paper 2 in terms of the users are participating and Paper 1 together with Paper 2 against Paper 3 by how the users are participating in the research setting compared to practice.

	Development	**Evaluation**
Research	Paper 1 - RQ1	Paper 2 - RQ2
Practice	Paper 3 - RQ3	

Table 4.1: Papers' area of research

We conducted a case study, contribution 1, with the purpose the end-user participation during development. In addition to this, a remote asynchronous usability test was conducted, contribution 2, involving the end-users. By including the end-users during development, we were able to gain knowledge about the context in which the application should be used. In addition to this, by showing the end-users the iterations of the application, we were able to adjust the application to better suit the end-user's actual needs. During the remote asynchronous usability testing, by giving the end-users the application, we were able to evaluate the application in the actual use context. This enable us to receive feedback, regarding the application, based on an actual integration of the application into the end-users' daily life.

Though the usability evaluation, contribution 2, we were able to asses the quality of the application, this included the utility, identified with the users though the development, and the usability of the application. The evaluation showed that the utility of the application fulfilled its purpose of helping the users record and manage their diabetes though blood glucose measurements, insulin calculations and meal planning. Furthermore, the usability evaluation showed that the flow of the difference functionalities was understood by the users, since only few usability problems concerned the inability to use or understand the flow of the application were identified. The majority of the usability problems concerned problems related to elements on individual pages.

From developing and evaluating the application we learned that the users had different roles as to how they were involved in the process, depending on if they were involved in development or evaluation. When the users were included in development they primarily had a participative, consultative and informative role as they were involved in meetings during

each sprint. This meant that they were supplying knowledge and ideas that then was used when designing and implementing the application. In addition, as the users were included in the development, their feedback and engagement were continuous and of a constructive and positive nature. When the users were evaluating the application they primarily had a participative and consultative role, as they performed the remote asynchronous usability testing where they reported usability problems and provided feedback though the UCI reports and diaries. Compared to development the user did not have an informative role, since this role is to provide information about the problem domain. When testing they gained hands-on experience with the system and by that could provide concrete feedback to the functionality of the application and the possible problems the application might have had. Some of the users evaluating the application had other expectations to the functionality of the application, which it was not able to deliver, than the users that participated during development. This had an impact on the engagement and type of feedback we received. The feedback received during development had a positive nature, this shifted during evaluation towards a more negative nature.

We performed a survey, contribution 3, researching how customer and end-user participation is performed in companies from the Danish software industry. When comparing these results, with the approaches used in contribution 1 and 2, we observed some interesting differences and similarities.

In the companies from the Danish software industry, some of the most used techniques for investigating the problem domain were Interviews and User stories. This is similar to the techniques used in contribution 1 to facilitate customer and end-user participation. Another observation was that for the companies, demonstrations were the most used technique during the Implementation- and Test phase, where in contribution 1 Walkthrough was the technique used. Despite these techniques being different there are still a similarity between them, this being presenting the user a prototype of the system. The technique used in contribution 2 was remote asynchronous usability testing, where for the companies Qualitative usability testing was the second highest voted technique used during the Test phase. Looking at each phase of development, we asked the respondents from the software industry in which phase they deemed customer and end-user participation most useful. Here the most prominent phase was Design followed closely by the Analysis-, Concept- and Test phase. This is similar to how the end-user was included in contribution 1 and 2 to great effect, where in contribution 1 the end-user was included in both the Concept-, Analysis- and Design phase. While in contribution 2 the end-user was included in the Test phase.

The primary obstacle answered by the companies was that customer and end-user participation was Lack of Time. This obstacle was confirmed in contribution 1, where each sprint had a specific set of user participatory activities, which had to be performed. This meant that scheduling the activities, in the time slots of a sprint, were difficult and time consuming. In addition to this, other techniques, such as observations that might had facilitated more user participation, were discarded because of the time factor. In contribution 2, concerning remote asynchronous usability evaluation, the obstacle of time were not experienced. One

of the benefits of remote usability evaluation is that it is less time consuming than regular laboratory- and field based usability testing, which also was the experience in contribution 2. Another high-ranking obstacle from the software companies was Geographical challenges. This was also experienced in contribution 1 and 2, where distance between the developers and the users was a dominant factor. In contribution 1 the problem of distance was alleviated by performing interviews using video conferencing and screen sharing. While in contribution 2, remote asynchronous usability testing was performed, thus eliminating distance and time as possible obstacles. Another obstacle from the software companies was customer and end-user participation Does not fit development process. The main focus of contribution 1 were to combine two development methods, to fit a high level of end-user participation, thus this obstacle was not experienced.

4.2 Limitations

We want to emphasize four limitations in our studies. First, our research is limited in terms of the number of UCI reports and diary entries received. Second, the research is limited by the number of users participating in the remote usability testing. Third, the development and evaluation performed in the research setting have been conducted by the authors and by that the effectiveness of this is also judged by the authors. Fourth, the limited scope of the survey investigating the practice of development and evaluation with customer and end-user participation.

4.3 Future works

We want to emphasize four possible future works. First, other possible participatory design methods should be combined with Scrum in order to get a more generalized understanding of Scrum combined with participatory design. Second, the effect of other techniques to facilitate customer and end-user participation during development and evaluation should be researched. Third, surveys investigating customer and end-user participation in practice should be performed, both to expand the scope outside of Denmark and to establish a deeper understanding of common practices of customer and end-user participation in the software industry. Fourth, research should be performed investigating how the different methods and techniques to facilitate customer and end-user participate are conducted in practice.

Bibliography

[1] Morten Sieker Andreasen, Henrik Villemann Nielsen, Simon Ormholt Schrøder, and Jan Stage. What Happened to Remote Usability Testing?: An Empirical Study of Three Methods. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1405–1414, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9. doi: 10.1145/1240624.1240838. URL http://doi.acm.org/10.1145/1240624.1240838.

[2] Morten Sieker Andreasen, Henrik Villemann Nielsen, Simon Ormholt Schrøder, and Jan Stage. What happened to remote usability testing? In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*, page 1405, New York, New York, USA, 2007. ACM Press. ISBN 9781595935939. doi: 10.1145/1240624.1240838. URL http://portal.acm.org/citation.cfm?doid=1240624.1240838.

[3] Dorine Andrews, Blair Nonnecke, and Jennifer Preece. Electronic Survey Methodology: A Case Study in Reaching Hard-to-Involve Internet Users. *International Journal of Human–Computer Interaction*, 16(2):185–210, 2003. doi: 10.1207/S15327590IJHC1602{_}04. URL https://doi.org/10.1207/S15327590IJHC1602_04.

[4] Ines Anić. Participatory Design: What it is, and what makes it so great. 11 2015. URL https://medium.com/@i_anic/participatory-design-what-it-is-and-what-makes-it-so-great-6bad619e8735.

[5] Isabelle Aujoulat, William DHoore, and Alain Deccache. Patient empowerment in theory and practice: Polysemy or cacophony? *Patient Education and Counseling*, 66(1):13–20, 4 2007. doi: 10.1016/J.PEC.2006.09.008. URL https://www.sciencedirect.com/science/article/pii/S0738399106003247.

[6] Muneera Bano and Didar Zowghi. User Involvement in Software Development and System Success: A Systematic Literature Review. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, EASE '13, pages 125–130, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1848-8.

doi: 10.1145/2460999.2461017. URL http://doi.acm.org/10.1145/2460999.2461017.

[7] Brenda Battleson, Austin Booth, and Jane Weintrop. Usability testing of an academic library Web site: a case study. *The Journal of Academic Librarianship*, 27(3):188–198, 5 2001. ISSN 0099-1333. doi: 10.1016/S0099-1333(01)00180-X. URL https://www.sciencedirect.com/science/article/pii/S009913330100180X.

[8] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for Agile Software Development. Technical report, 2001. URL http://agilemanifesto.org/.

[9] Scott Carter and Jennifer Mankoff. When participants do the capturing: The role of media in diary studies. pages 899–908, 2005. doi: 10.1145/1054972.1055098.

[10] CollabNet VersionOne. DevOps and Agile Solutions for the Enterprise | CollabNet VersionOne, 2019. URL https://www.collab.net/.

[11] Natalie J Corrao, Alan G Robinson, Michael A Swiernik, and Arash Naeim. Importance of Testing for Usability When Selecting and Implementing an Electronic Health or Medical Record System. *Journal of Oncology Practice*, 6(3):120–124, 5 2010. ISSN 1554-7477. doi: 10.1200/JOP.200017. URL https://doi.org/10.1200/JOP.200017.

[12] MICK P COUPER. Review: Web Surveys: A Review of Issues and Approaches. *Public Opinion Quarterly*, 64(4):464–494, 2000. ISSN 0033-362X. doi: 10.1086/318641. URL https://doi.org/10.1086/318641.

[13] Mick P Couper, Arie Kapteyn, Matthias Schonlau, and Joachim Winter. Noncoverage and nonresponse in an Internet survey. *Social Science Research*, 36(1):131–148, 2007. ISSN 0049-089X. doi: https://doi.org/10.1016/j.ssresearch.2005.10.002. URL http://www.sciencedirect.com/science/article/pii/S0049089X05000724.

[14] WJ Doll and G Torkzadeh. The measurement of end-user software involvement. *Omega*, 18(4):399–406, 1 1990. ISSN 0305-0483. doi: 10.1016/0305-0483(90)90030-D. URL https://www.sciencedirect.com/science/article/pii/030504839090030D.

[15] Susan Dray and David Siegel. Remote Possibilities?: International Usability Testing at a Distance. *Interactions*, 11(2):10–17, 2004. ISSN 1072-5520. doi: 10.1145/971258.971264. URL http://doi.acm.org/10.1145/971258.971264.

[16] Kathleen M Eisenhardt. Building Theories from Case Study Research. *The Academy of Management Review*, 14(4):532–550, 1989. ISSN 03637425. doi: 10.2307/258557. URL http://www.jstor.org/stable/258557.

[17] Weimiao Fan and Zheng Yan. Factors affecting response rates of the web survey: A systematic review. *Computers in Human Behavior*, 26(2):132–139, 2010. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2009.10.015. URL http://www.sciencedirect.com/science/article/pii/S0747563209001708.

[18] Rashina Hoda, James Noble, and Stuart Marshall. The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology*, 53(5):521–534, 2011. ISSN 0950-5849. doi: https://doi.org/10.1016/j.infsof.2010.10.009. URL http://www.sciencedirect.com/science/article/pii/S0950584910001941.

[19] Jakob Nymann Holgersen, Mathias Huse Jensen, and Pelle Ulmer Jørgensen. Combining participatory design into an agile development method. Technical report, Aalborg University, Aalborg, 2018. URL https://projekter.aau.dk/projekter/da/studentthesis/combining-participatory-design-into-an-agile-development-method(1f91831c-a09c-4be3-959b-57bf9f443257).html?fbclid=IwAR29_t7ApVRAInEkzqJEflj-mJIi-mWPmKTObQPB2MAJ_3Il9gbx2J4ycaUJEflj-mJIi-mWPmKTObQ.

[20] Kyungdoh Kim, Robert W Proctor, and Gavriel Salvendy. The relation between usability and product success in cell phones. *Behaviour & Information Technology*, 31(10): 969–982, 10 2012. ISSN 0144-929X. doi: 10.1080/0144929X.2011.596997. URL https://doi.org/10.1080/0144929X.2011.596997.

[21] S Kvale. *Interviews: an introduction to qualitative research interviewing*. Sage Publications, 1996. ISBN 9780803958197. URL https://books.google.dk/books?id=tJPZAAAAMAAJ.

[22] Paul Lavrakas. Longitudinal Studies, 2008. URL https://methods.sagepub.com/reference/encyclopedia-of-survey-research-methods.

[23] Jonathan Lazar, Jinjuan H Feng, and Harry Hochheiser. *Research methods in human-computer interaction*. 2017. ISBN 9780128053904. URL http://www.worldcat.org/isbn/9780128053904.

[24] James R Lewis. Sample Sizes for Usability Tests: Mostly Math, Not Magic. *Interactions*, 13(6):29–33, 11 2006. ISSN 1072-5520. doi: 10.1145/1167948.1167973. URL http://doi.acm.org/10.1145/1167948.1167973.

[25] José Luis Monteagudo Peña and Oscar Moreno Gil. *e-Health for Patient Empowerment in Europe*. 10 2007. ISBN 978-84-95463-56-2.

[26] Jennifer Mankoff. Crossing Qualitative and Quantitative Evaluation in the Domain of Ubiquitous Computing. Presented at the CHI 2005 Workshop on Usage Analysis: Combining logging and qualitative methods. 2005.

[27] A. Martin, R. Biddle, and J. Noble. The XP Customer Role in Practice: Three Studies. In *Agile Development Conference*, pages 42–54. IEEE. ISBN 0-7695-2248-3. doi: 10.1109/ADEVC.2004.23. URL http://ieeexplore.ieee.org/document/1359794/.

[28] Hendrik Müller, Aaron Sedley, and Elizabeth Ferrall-Nunge. Survey Research in HCI. In Judith S Olson and Wendy A Kellogg, editors, *Ways of Knowing in HCI*, pages 229–266. Springer New York, New York, NY, 2014. ISBN 978-1-4939-0378-8. doi: 10.1007/978-1-4939-0378-8{_}10. URL https://doi.org/10.1007/978-1-4939-0378-8_10.

[29] John Murphy, Steve Howard, Jesper Kjeldskov, and Steve Goschnick. Location, location, location: Challenges of Outsourced Usability Evaluation. 2004.

[30] Leysia Palen and Marilyn Salzman. Voice-mail Diary Studies for Naturalistic Data Capture Under Mobile Conditions. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, CSCW '02, pages 87–95, New York, NY, USA, 2002. ACM. ISBN 1-58113-560-2. doi: 10.1145/587078.587092. URL http://doi.acm.org/10.1145/587078.587092.

[31] Prof. Dr. Werner Patzelt. Case Study Research and the Grounded Theory Approach, 2019. URL http://ipsa.sbe.metu.edu.tr/case-study-research-and-grounded-theory-approach.

[32] Roman Pichler. *Agile Product Management with Scrum: Creating Products That Customers Love*. Addison-Wesley Professional, 1st edition, 2010. ISBN 0321605780, 9780321605788.

[33] H. Robinson and H. Sharp. XP culture: why the twelve practices both are and are not the most significant thing. In *Proceedings of the Agile Development Conference, 2003. ADC 2003*, pages 12–21. IEEE, 2003. ISBN 0-7695-2013-8. doi: 10.1109/ADC.2003.1231448. URL http://ieeexplore.ieee.org/document/1231448/.

[34] Hugh Robinson and Helen Sharp. The Social Side of Technical Practices. In Hubert Baumeister, Michele Marchesi, and Mike Holcombe, editors, *Extreme Programming and Agile Processes in Software Engineering*, pages 100–108, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31487-5.

[35] O Sohaib and K Khan. Integrating usability engineering and agile software development: A literature review. In *2010 International Conference On Computer Design and Applications*, volume 2, pages 2–32, 2010. doi: 10.1109/ICCDA.2010.5540916.

[36] Hrafnhildur Sif Sverrisdottir, Helgi Thor Ingason, and Haukur Ingi Jonasson. The Role of the Product Owner in Scrum-comparison between Theory and Practices. *Procedia - Social and Behavioral Sciences*, 119:257–267, 3 2014. ISSN 1877-0428. doi: 10.1016/J.SBSPRO.2014.03.030. URL https://www.sciencedirect.com/science/article/pii/S1877042814021211.

[37] D A Vaus. *Surveys in social research*. Routledge, Abingdon, Oxon, 2014. ISBN 978-0-415-53015-6.

[38] Robert K Yin. *Case Study Research and Applications: Design and Methods - Robert K. Yin - Google Books*. SAGE Publications, sixth edition edition, 2017. ISBN 9781506336169.

A | Papers

A.1 Combining Scrum with participatory design: A case study of end-user participation

(1) Jakob Nymann Holgersen, Mathias Huse Jensen, Pelle Ulmer Jørgensen. Combining Scrum with participatory design: A case study of end-user participation. *Department of Computer Science, Aalborg University*. Aalborg 2019.

A.2 A longitudinal evaluation of a mobile application using two remote asynchronous usability test methods

(2) Jakob Nymann Holgersen, Mathias Huse Jensen, Pelle Ulmer Jørgensen. A longitudinal evaluation of a mobile application using two remote asynchronous usability test methods. *Department of Computer Science, Aalborg University*. Aalborg 2019.

A.3 Customer and end-user participation in software development: A survey of the Danish software industry

(3) Jakob Nymann Holgersen, Mathias Huse Jensen, Pelle Ulmer Jørgensen. Customer and end-user participation in software development: A survey of the Danish software industry. *Department of Computer Science, Aalborg University*. Aalborg 2019.

Combining Scrum with participatory design: A case study of end-user participation

Jakob Nymann Holgersen
Department of Computer Science
Aalborg University, Denmark
jholge14@student.aau.dk

Mathias Huse Jensen
Department of Computer Science
Aalborg University, Denmark
matjen13@student.aau.dk

Pelle Ulmer Jørgensen
Department of Computer Science
Aalborg University, Denmark
pjarge14@student.aau.dk

Abstract

The essence of Scrum is limited documentation, fast releases and adaptability to changes. The responsibility of a Product Owner is to ensure the needs of the end-user. Studies have shown that customer representative only have a limited understanding of users' needs, which means that there is a need to adapt the existing method to further ensure end-user participation. This paper presents a case study where Scrum was combined with a participatory design method to further ensure end-user participation in every step of development, except the implementation activities. This study documents the experiences of ensuring end-user participation during development of a mobile application for diabetics. The findings show, that the adapted development method facilitates a high level of end-user participation throughout the entire development process, except the implementation activities.

Keywords Combining development methods, System development, Scrum, MUST, Agile development, Participatory design, End-user, Case study, Diary study, Patient empowerment

Introduction

A study from 2017 of 1492 software companies worldwide, show the most common reasons for adapting to an agile development method. This includes the ability to adapt to changing priorities, as well as increase productivity and by that accelerate software delivery [5]. 70% used a Scrum based development method (Scrum, ScrumBan and Scrum/XP Hybrid). 14% used a Hybrid of multiple methodologies, 5% used Kanban where Lean, XP, Iterative Development and the Spotify model constitutes the last 5% [5].

Scrum is an agile development framework, that is built around three central concepts that are, transparency, inspection and adaptation [21]. Transparency means that the process must be visible and understood by everyone involved. To facilitate transparency Scrum uses different Artifacts, these being Product Backlog, Sprint Backlog and Increments. Inspection means that all Artifacts in the process should be inspected frequently to detect undesirable variances. Adaptation means that the process should be able to adapt to changes so if an aspect deviates, the process must be able to adapt to this.

Scrum has four events for inspection and adaptation, these being Scrum Planning, Daily Scrum, Sprint Review and Sprint Retrospective. Each of these events is a part of a fifth event [21]. The fifth event is the Sprints, which are iterative and of a fixed length. A team in Scrum revolves around three roles, a Scrum Master, a Product Owner and the Development Team.

The Product Owner has the sole responsibility for managing the Product Backlog. They are responsible for providing guidance and support towards the project's success. Hence, being a Product Owner is usually a full-time job [15][24]. The role of the Product Owner is often a representative of the customer ordering the system.

Studies have shown that the customer representative only have a limited understanding of the users' needs, and that they might not be the actual users of the system [13][18][19]. While Scrum has clear techniques of working with the Product Owner, there exist no techniques to ensure inclusion of the end-user [3][23].

Participatory design is an entirely different approach where there are active end-user participation throughout the development. The goal of participatory design is to involve end-users, stakeholders, designers and researchers in the design process, such that the product is created to satisfy the needs of its user-base [1][14]. MUST is a well-documented participatory design method, that is the result of a ten-year research collaboration with a series of Danish and American companies. MUST focuses on developing IT-solutions that account for the different qualifications and skills of employees. Furthermore, it is described by the authors as a method that is based on thorough participation with users and managers [8][22]. In addition, Kujala [9] describes MUST as a method that combines the use of ethnographic techniques and intervention, within the participatory design tradition, in the context of custom development using design processes

such as contextual design. The purpose of MUST is to provide guidelines and techniques on end-user participation and result in an IT design project. An IT design project forms the foundation for further decisions regarding choosing the most appropriate IT solution [8][22]. MUST consists of four different phases, these being Initiation, Strategy, Analysis and Innovation.

This paper presents a case study where Scrum was combined with a participatory design method, MUST, to further ensure end-user participation in every step of development, except the implementation activities. The study documents the developers experiences with end-user participation during development of a mobile application for diabetics, with a focus on patient empowerment. Patient empowerment refers to the philosophy of giving the ability to act and take informed choices regarding health treatments back to the patient [2][12]. The feasibility of combining Scrum and MUST is based on whether it allows and facilitates active end-user participation throughout the entire development, except the implementation activities.

In the following section we describe previous research on combining agile development with participatory design. Following this we present a proposal as to how Scrum can be combined with MUST. Then the findings are described and a discussion of a refined adapted development method is made based on the experiences learned during development. Lastly, the practical reality of the adapted development method and unexpected results are discussed.

Related work

In the following section, we will present an overview of relevant literature, with regards to agile development and end-user participation in the actual development process.

Case studies

Kautz [7] shows a case study where techniques for user participation is included in agile development. The framework used, to include users, is based on participatory design litterateur. The framework is included in different stages during the agile development: The preparation stage, the development stage and the evaluation stage. The end-users' are involved as an on-site customer. The end-user assumed different roles during the different stages of development. In the preparation stage the end-user's roles are participatory and informative where in the evaluation stage the roles are participatory and consulting. The end product is deemed successful by both the developers and the customer, as this type of user participation supported a balance between project progress and flexibility. Kautz [7] observed difficulties when working with end-users regarding their understanding of what is possible. An example is that when a manager proposed two different ways to have the same functionality, the customer just said yes to both without having the understanding of the functionality being mutually exclusive.

Hansson et al. [6] present a case study the describes a case where participatory design techniques are integrated into an agile development method. The case revolves around a company that develops a booking system. The participatory design techniques that are utilized are performed remotely since the users of the system are spread across multiple countries. The techniques used are user-meetings, support services and courses. Hansson et al. [6] discovered, that by using participatory techniques, the company is able to obtain more and better knowledge about the users than when only using agile methods. Furthermore, Hansson et al. [6] discovered that, the use context of the system is often not understood properly, if only analyzed using agile techniques. Hansson et al. [6] concludes that a combination of participatory design and agile development could result in a more comprehensive system that are better suited to the users' needs and are better able to fit the appropriate use context. It was discovered that not all the different users are participating, so when new changes are made, these changes are only based on the participating users. Furthermore, the prioritization of functionality is controlled entirely by the developers, meaning that the end-users have no impact on this process.

Rittenbruch et al. [17] provide a comparison of the conceptual and methodological similarities of Extreme Programming (XP) and participatory design. On this basis, they set out to develop an approach, which integrates techniques from both agile development and participatory design. In order to test their method, they performed a proof of concept research project, with IT researchers as customers. This research provides a series of recommendation to overcome the conception of limited user participation in XP. However, they do not provide any observation concerning the practical application with regards to user participation.

Frameworks

Chamberlain et al. [4] attempts to develop a framework for integrating agile development and user-centered design. The approach is to analyze similarities and differences between agile development and user-centered design. By this analysis they conclude that these two frameworks are compatible. However, a practice study shows that there is a clear distinction between designers, who are responsible for the user-centered activities, and the developers. Furthermore,

the practice study shows little customer and user participation during the development. The user participation is through interviews, where the users are asked for their opinion of the system, they are shown during a short test session. Chamberlain et al. proposes five principles for a successful integration of the two frameworks. These being user involvement, collaboration, prototyping, project life cycle and project management. The study does not provide empirical data to validate these five principles and by that the validity of the proposed framework [4].

Preece and Rombach [16] describe a Software Engineering framework and a Human Computer Interaction framework. The article suggests a combination of the two frameworks by investigating what identifies each framework and what is in common between the frameworks. The purpose is to be able to characterize Software Engineering and Human Computer Interaction measurements to provide a common understanding of these [16]. They conclude, that their findings are able to bring the two frameworks closer together, as they provide a common understanding of the different characterizations of each framework through their own framework [16]. By using examples from their own work, they show that their framework is able to facilitate four key functions, these being: post hoc analyses, planning aid, guiding practice and to facilitate communication. However, none of the proposed examples take place in practice, hence they are not able to give an indication of if their framework could work in practice.

Litteratur reviews

Salah et al. [20] conducts a literature review which aims to identify various challenging factors that restrict the integration between agile development and user-centered design. One factor identified is the lack of time for upfront activities such as researching the user and their work practices. The identified practice to accommodate this, is to include an upfront design phase [20]. Another factor identified is that it is problematic to prioritize user-centered design activities. The identified solution to this is to create an individual backlog for all user-centered activities and to have a designated UX (User experience) person being responsible for the backlog [20]. Another discovered challenge is to synchronize the work of the developers and the user-centered design practitioners. The solution to this, is that the UX team are part of the Daily Scrum thus the UX team are better able to communicate and coordinate with the development team [20].

Larusdottir et al. [10] conducts a synthesis paper that integrates results from interviews and survey studies related to the research of using User-centered systems design activities in agile projects in practice. A total of 37 IT professionals were interviewed and further 158 professionals contributed with data on the surveys [10]. These results are analyzed by the authors and the four principles of the agile manifest is discussed. From the analyze they discover that the principle "Responding to change" was highly prioritized when categorized as a defect reported by a user, but when the change was on the form of design changes reported by usability experts or users, this was less prioritized. Furthermore, from the interviews they discover that some of the agile development processes neglects the early phases and believes that this can be accounted for in a later phase [10]. Larusdottir et al. [10] suggest that user experience professionals get more authority and the ability to stop the production line if the current design is not perceived as useful by the users.

Development method

Through the previous described related work, integrating participatory design into agile development is done by choosing participatory techniques that are then fitted into the agile development cycle. This research aims to integrate the agile development method, Scrum, with the participatory design method, MUST, by integrating the MUST method and its phases with Scrum.

Through the four phases of MUST, different techniques can be utilized to include the end-user in development and thereby obtain important feedback that reflects the end-users needs. The nature of these phases is in a waterfall structure, hence by combining it with Scrums iterative characteristics, it is possible to use the techniques from MUST's phases in Scrum's sprints and planning. By doing this the aim is to include the end-user, by generating requirements, getting feedback on functionality and new ideas in every step of the development, except the implementation activities.

Combining Scrum and MUST

The development, in this research, is initiated with a phase called the pre-sprint. The purpose of the pre-sprint is to identify the problem that the case revolves around. Furthermore, the pre-sprint is used to gain knowledge about the different concepts involved in the case. The Initiation phase from MUST is utilized for this. The purpose of the Initiation phase is to gather knowledge, to help create concrete tasks that need to be solved. Another purpose of the Initiation phase is to plan the project. The reason, for using the Initiation phase before beginning the Scrum sprints, is that planning the project before implementation starts is required. The Initiation phase is able to assist with this, with techniques such as Baseline Planning. The result of the Initiation phase is a

Project charter, that outlines the project and the techniques that are going to be used throughout the project.

Each Sprint is planned as a normal Scrum sprint, where Sprint goals are constructed. The activities performed during the Sprints are different from regular Scrum, since a Sprint is not focused solely on development but also have information gathering techniques included. The first part of each Sprint is focused on the In-line- and In-depth analysis phases from MUST. The purpose of the In-line analysis phase is to investigate the feasibility of the proposed solution. This is done through a SWOT analysis. Furthermore, a Functional analysis is constructed to gain further knowledge about the domain the application is part of. The purpose of the In-depth analysis phase is to get an understanding of what and how the product should be designed. To facilitate this, interviews with the primary stakeholders are conducted. In addition to this, meetings with the Product Owner discussing possible functionality are held. Furthermore, Walkthrough are conducted with the primary stakeholders, where the prototype application is demonstrated. The purpose of the Walkthrough is to not only validate the constructed prototype but to also generate additional knowledge based on the current functionality. The result of the In-line analysis phase is a Strategy report outlining the surroundings and work area that need to be considered regarding the application. The In-depth analysis phase produces an Analysis report, outlining the goals, needs and problems the primary stakeholders experience regarding the domain.

The last phase, constituting the last part of each sprint, is the Innovation phase, where the application is developed based on the Analysis report. The Innovation phase starts with a meeting with the Product Owner, where the Product Backlog is prioritized, and the Sprint Backlog is constructed. In the Innovation phase requirements are constructed based on the goals, needs and problems from the Analysis report. Furthermore, the Sprint Backlog is implemented during the Innovation phase.

Case study

We have conducted a case study using an agile development method combined with a participatory design method, developing an Android application supporting patient empowerment. The entire development of this study is split into three Sprints that each last three weeks. An overview of the three Sprints and how MUST and Scrum are combined can be seen on Figure 1. The results and experiences of applying our adapted development method was collected through the use of Interviews, Walkthrough and documentation of the development through diary entries.

Participants: 6 participants were involved in the project, 5 as end-users and 1 as Product Owner. 3 type 1 diabetics, all female, and 3 type 2 diabetics 2 male and 1 female. Table 1 shows the demographics of the participants and our Product Owner. The participants were found with the help of the Local Diabetes Association.

The authors constitutes the development team with shared responsibility of acting as Scrum Master. The Scrum Master role was shared by having all members being responsible for and helping with planning the Sprints, managing backlogs and ensure that goals were met. Documentation and entries in the diary were written by the authors.

Gender (age)	**Participants**		
	Type of diabetes	**Time with diabetes**	**Treatment**
Product Owner			
Male (61)	Type 2	15 years	Pen
Participants			
Male (51)	Type 2	23 years	Pen & tablets
Female (57)	Type 2	5 years	Pen & tablets
Female (28)	Type 1	22 years	Pump
Female (36)	Type 1	4 weeks	Pen
Female (51)	Type 1	42 years	Pump

Table 1. Product Owner and participants demographics

Setting: This research was conducted as a master thesis meaning that the development of the application is conducted in a university. This meant that 1/3 of the work-hours was allocated to university courses. This had an impact on planning and the allocated time for techniques used to ensure end-user participation.

System: The materials used for this research was the Android application developed using an adapted development method to include end-users into the development process, except the implementation activities. The goal of the developed application was to support diabetic in their daily life, by providing custom management tools and information in a single solution.

Procedure: The procedure of development was following the previous described combination of Scrum and MUST. All walkthroughs with end-users were conducted by one of the authors, where the end-user was guided through all new developed functionality. Following the walkthrough, a semi-structured interview was conducted regarding the functionality shown during the walkthrough.

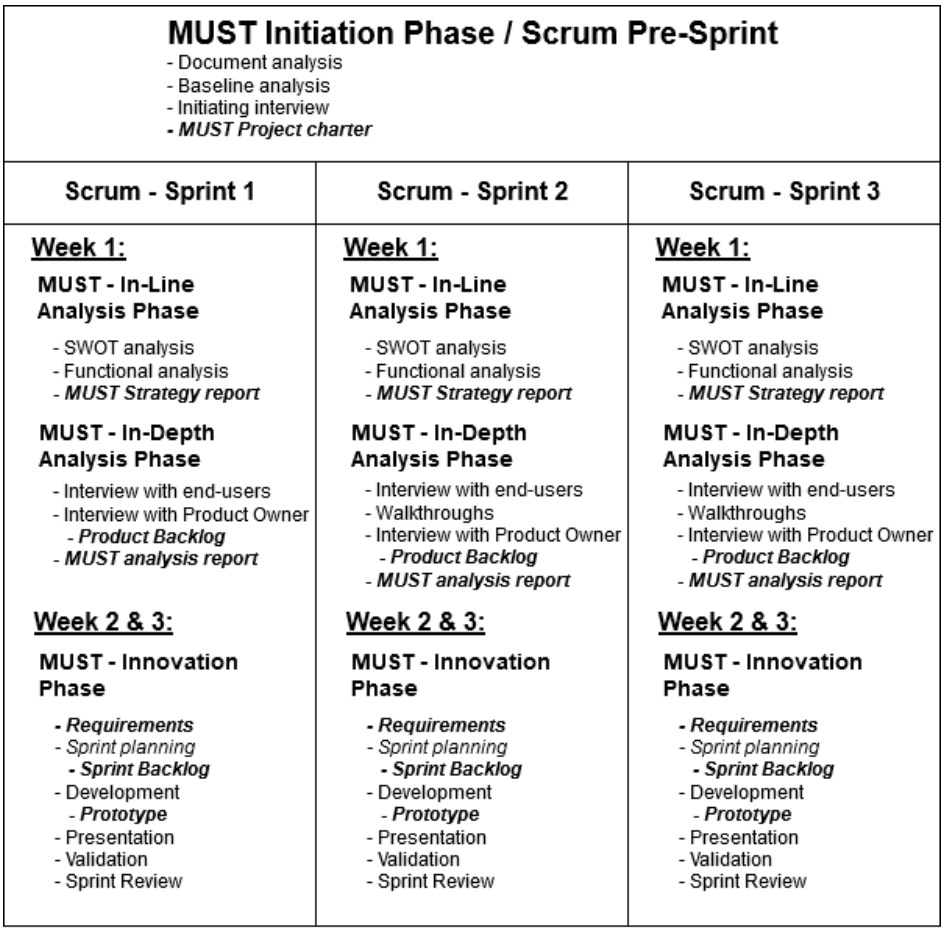

Figure 1. Combination of Scrum and MUST

Data collection: The primary data collection was the diary used to document observations made during the development process. Furthermore, secondary data collection was through interviews and walkthroughs of the application. The interviews and walkthroughs were conducted with the diabetics to get further insight into the effectiveness of the adapted development method. The diary documents the use of the adapted development method.

Data analysis: Each diary entry was analyzed and notable entries were marked. The notable entries were then further analyzed to find the entries that best documents the experiences of end-user participation during development. Furthermore, the interviews and walkthroughs were analyzed to further identify how the end-user participation was performed and its effect on the development of the application.

Findings

In the following the experiences made during the combination of the participatory design method MUST and the agile development method Scrum, will be described through the use of quotes from the diary, made during development.

End-user participation as preparation before the first Scrum sprint

The Initiation phase, from MUST, provided an understanding of the various aspect and criteria for the success of the project, before the beginning of the first sprint of Scrum. The Initiation phase allowed to plan the project and enabled the development team to get early knowledge of the problem domain in the form of a Document analysis.

The Initiation phase included initiating interviews with a diabetic (end-user) and a chief physician to further gain knowledge about the problem domain, as stated in the diary:

"*The interview was conducted as part of the Initiation phase to gain knowledge about what it is like to live with diabetes*". This knowledge helped with not only identifying problems but also gained an understanding of the different aspects of a diabetic's life.

The Baseline planning technique was used in combination with Sprint Planning to plan the project. Combining Baseline planning and Sprint Planning gave a well-defined structure on not only how long each Sprint should be, but also what a Sprint should contain. These two techniques are complementary, when structuring the project, as Baseline planning allowed the developers to outline the entire project and Sprint Planning allowed for for detailed planning of the individual Sprints.

Validation of implemented requirements with end-users

One of the techniques utilized from the MUST Innovation phase was Walkthrough. By using Walkthrough, it was possible to confirm if the functionality arising from the requirements, discovered during the In-depth analysis phase interviews, was what the end-user had in mind. As mentioned in the diary, we discovered that: "*Furthermore, the walkthrough would function as validation for Sprint 1, since we would get confirmation regarding the implemented functionality of the prototype*". This validation showed, if the utility provided by the application, matched the expectations of the end-users.

Generating new requirements iteratively using end-user participation

From the walkthroughs that were used in the end of the Innovation phase, to validate developed functionalities, we discovered that the new functionalities the end-user were presented, spiked new ideas in the end-user. As mentioned in the diary entry from Sprint 3 where: "*Both type 2 interviewees mentioned that they would like a feature to export their blood glucose measurements*". The type 2 diabetics did not previously mention the idea of exporting their blood glucose measurements, but as they were presented to functionality where they could enter and save their blood glucose measurements, they got the idea of exporting it. Another example of implemented functionality spiking new ideas, can be seen from a follow-up interview with a type 1 diabetic: *"Interviewee 2 mentioned that on the meal log, which displays already calculated meals, it would be beneficial, if you were able to change the blood glucose level, but use the same products"*. The initial idea of having a meal log did not include the option of changing the blood glucose level. However, here the walkthrough spiked an idea of expanding the implemented functionality.

Iteratively analyzing the environment of the project and the end-users

From constructing a Strategy report in each Sprint, we learned that the SWOT analysis helped to determine possible weaknesses and threats to the project. During development, it was discovered that, performing a SWOT analysis for each Sprint was excessive. As the Sprints only lasted three weeks, the SWOT analysis only had minor changes, as mentioned in the diary: "*At the start of the second Sprint most of these factors have not changed. However, patient empowerment can now be identified as Strengths for the project, as aspects supporting patient empowerment have been implemented*". This led us to decide, that for the third Sprint performing a SWOT analysis was not useful.

We experienced that by conducting a Functional analysis as part of the In-line analysis phase iterativly contributed to the development. The revision of the Functional analysis resulted in new findings within all three levels of the Functional analysis. For example, it was discovered that the surrounds of the type 2 diabetics were different than those of a type 1. For an example, type 2 diabetics uses their general practitioner while type 1 uses a diabetes physician. This was described in the diary: "*It was discovered that type 1 diabetics have more direct contact with the diabetes department on the hospital, while type 2 diabetics primarily have contact with their general practitioner instead of the hospital*". Revising the bottom level of the Functional analysis contributed with new knowledge, that helped in identifying new functionality.

It was mentioned in the diary that: *"In Sprint 3, we decided not to review the SWOT analysis, as the factors identified in Sprint 1 and Sprint 2 are not excepted to change in the three weeks since the latest review of the SWOT analysis. This is also the case for the Functional analysis, as the surroundings and possibilities in the top level are likely not to change in the short time span of this project"*. This showed that for this project iterating the Strategy report each three weeks, contributed little to the development, since the changes discovered only had a minor effect on the functionality of the application and therefore deemed not useful to do iterativly. Furthermore, the changes to the levels in the Functional analysis can be contributed to the interviews with type 2 diabetics, which were not conducted in Sprint 1 as no contact was established at that time.

Constructing Analysis reports iteratively using end-user participation

When constructing the Analysis report we experience that, during the In-depth analysis phase, scheduling the interviews proved to be problematic, as mentioned in the diary: *"Coordinating interviews and analyzing these have been a challenge since there can be wasted time between the analysis of one interview and conducting the following interview. Furthermore, we have finished all the remaining scheduled interviews and are waiting for the rescheduling of the next interview which was caused by a cancellation"*. The downtime between the interviews were used to analyze the already conducted interviews. It was discovered that the goals, needs and problems outlined in the Analysis report, were easily translatable into items for the Scrum Product Backlog, as written in the diary: *"The Product Backlog itself is constructed based on the Analysis' reports goals, needs and problems section. This section is easily translatable into the Product Backlog since each item reflect a possible functionality of the prototype"*. The easily constructed Product Backlog made the transition from analysis to implementation fast and effective during the project.

Changing the adapted development method

Through the use of the adapted development method, we experienced challenges and room for improvement. These challenges were mainly posed by the iterations of activities in the In-line analysis phase. Despite the importance of investigating the environment of the project through a SWOT analysis, and the end-users through a Functional analysis, it was discovered that iterating these analyses, does not contribute sufficient with new knowledge given the time spend. Furthermore, giving the time-slot of one week allocated for iterating the SWOT- and Functional analysis as well as information gathering through interviews were a challenging task. This meant that information gathering often consumed more time than allocated for the task.

In the initial combination of Scrum and MUST, the first Sprint was mainly used for initial information gathering through interviews. We did not include Walkthrough in the end of the Pre-Sprint nor the beginning of the first Sprint and by that we had no feedback on the initial design. This meant that the first week of the first Sprint was further information gathering through interviews which resulted in ideas for requirements. This meant that despite the Pre-Sprint and MUST Initiation phase, we did not have sufficient knowledge about what the end-users' wanted to start designing, hence the first week of the first Sprint functioned as an extension to the Pre-sprint.

To accommodate for these challenges, an improved adaptation to the adapted development method is proposed. Before starting the sprints, a full cycle of MUST phases is worked through. The idea behind this is to start the project out with extensive planning and information gathering, which is the focus of the first 3 phases of MUST: Initiation, In-line analysis and In-depth analysis. Where this information should be used to make a mock-up of the initial design, in the Implementation phase, which can be used to generate new ideas in the first Sprint. Going into the Sprints, the changes made here, is aimed at increasing the techniques used for the initial information gathering, such as Interview, Observation, Walkthrough and Think-aloud. The goal for the first week is to produce a prioritized Product Backlog and documentation being on the MUST analysis report form. The remaining two weeks of the Sprint is allocated for the development. The goal for these two weeks is to develop an increment of the working prototype, which will be used to validate and generate new ideas to functionality. These evaluations is fluent across the end of the Sprint and the start of the next Sprint's first week, as these evaluations serves as both the validation and the collection of new information regarding possible improvements and features.

The effect of end-user participation on the development team

Actively including and working with the end-users through the entire development, except in implementation activities, had an effect on the developers' mindset. This caused the developers to start thinking differently, as they were able to talk with the end-users more often and by that have a better idea of what to implement. Furthermore, by having the end-users participate in creating backlog item and then validating the implemented items, the developers experienced a better connection with and understanding of the end-users. This mindset contributed to the developer being more open and willing to facilitate the end-user participation. Both in terms of the time slot allocated for the participatory design techniques, but also seeing the importance of these techniques and not only focusing on the technical tasks related to implementation.

Discussion

This section discusses interesting observations made during the research. Furthermore, this section discusses the findings in relation to related work. This is followed by the practical implication of combining an Scrum with a participatory design method.

An interesting observation was that besides the confirmation of the implemented functionality and spike of new ideas, the walkthroughs and follow-up interviews revealed differences that need to be considered regarding the diabetes type. An example of this was from an interview with a type 2 diabetic. Here we discovered that their medical contact is different than type 1, as mentioned in the diary: "*Both interviewees mention when asked about contacting a professional regarding questions, that they had to contact their own general practitioner*". This showed that there were differences in regards to who the diabetics could contact about questions regarding their condition. For a type 1 diabetic there exist a specific ward, where type 1 diabetics go to consultation, while type 2 diabetics go to their own general practitioner. This has an effect on the level of guidance the diabetics are provided, which has an impact on the level of supporting information each diabetic need.

Furthermore, we discovered that Diabetes is a complex condition but that there are many more factors effecting what functionality an end-user wants. An example of this was when prioritizing the parental supervision functionality. Here one type 1 diabetic prioritized it as the most important functionality while two other type 1 diabetics prioritized it as the least important. This indicates that the individual's situation have a significant effect on the desired functionality.

Another interesting observation was that the feedback received through the walkthroughs, further contributed with motivation towards continues improvement and development. At the walkthroughs we observed that the end-user was either positive about the features and complemented the application and thereby our work or highly interested in discussing improvements. This was motivating and gave a desire to keep working on the application, as we felt that the work and by that the application was wanted by the users.

In relation to related work, Kautz [7] showed additional considerations that can be made in addition to our adapted development method. The different roles, expressed by the on-site customer in the article, can be related to the different phases outlined in our adapted development method. In our approach the Pre-Sprint is focused on information gathering. Kautz [7] describes the role for the on-site customer during analysis as participatory and informative. It would be possible to further plan our adapted development method around the different roles of the participants, meaning that techniques should be chosen to fit the different roles of the participants during development.

Hansson et al. [6] showed similar findings to ours, since they showed that non-traditional participatory design activities were beneficial when generating new ideas and functionality for a system. Hansson et al. [6] showed that development using pure agile methods resulted in a lack of knowledge about the end-user and the use context of the system. This observation is further backed by Larusdottir et al. [10], which discovered that agile development tend to neglect the upfront activities. Furthermore, as shown by Salah et al. [20] the literature describes a lack of time for upfront activities, where they identified practices, where one is to include an upfront design phase. Hanson et al. [6] showed that by including participatory techniques, it allowed the developers to acquire knowledge about the users and account for development issues. This is similar to what we discovered during our research, as we showed that new requirements were identified by the end-user throughout the entire development and that we were able to account for these during development. Furthermore, we propose that having a full cycle of MUST before starting the development Sprints could be beneficial to create a better knowledge base. In the case study made by Hansson et al. [6] the prioritization of functionality is still controlled by the developers. For our case study, the developers also had the final say regarding what functionality would be implemented. All end-users were asked to make their own prioritization so we would better be able to ensure that the correct functionality were created, in addition to the prioritization by the Product Owner.

In relation to practice, this paper presented a combination of two development methods to better include the end-users, throughout the entire development process. Furthermore, adaptations to the adapted development method have been discussed in order to mitigate and improve the combination based on our experience. Following the findings of this research and the discussion of this, the practical reality is that an initial phase is required before implementation starts. This initial phase is needed to not only establish contact with the end-users but to also gain information about them and their environment. Another practical reality of this research is that it is needed to collect additional information from the end-users between each implementation iteration, where in this research it is done through Walkthrough and follow-up Interview. This is needed since the end-users might be able to provide additional knowledge or identify additional problems based in the already implemented functionality. Another practical reality is that the developers need to be present at the techniques used to ensure end-user participation. This research showed that by having the developers interacting with the end-users, the developers gain an understanding of the end-users and their environment. Furthermore, by having the developers meeting with the end-users the developers are better able to ask clarifying questions to ensure that the correct functionality is implemented.

Conclusion

End-user participation in agile development is often limited to the Product Owner, where the needs of the end-users is determined by the Product Owner. This paper presented a case study investigating the use of an adapted development method to better ensure end-user participation in the development process. This paper presented findings regarding employment of an adapted development method, in a case concerning development of a mobile application for diabetics. The findings show that we experienced a high level of end-user participation throughout the entire development process, with the exception of implementation activities. Through development we learned that some of the activities chosen were more important than others when these were part of an iteration. For example, we learned that iterating a SWOT analysis every Sprint was unnecessary, where iterating Walkthrough and follow-up Interview were critical for the development. Furthermore, it was shown that by having end-user participation we were able to continuously validate the implemented utility and to further generate new requirements for the system.

To validate our research we conduct our case study according to [11], in which we attempt to mitigate weaknesses of a case study. A challenge with case studies is to find the correct participants, where for this paper both type 1 and type 2 diabetics is chosen to get a broader perspective of diabetes. The diary is conducted by us, the authors and developers, meaning that we are able to account better for some of the difficulties of using a diary, since we are able to guarantee that entries are made every day. Furthermore, we are able to outline specifically what needs to be described in the diary and in what level of detail [11]. We acknowledge that we are not able to make generalized conclusions regarding the usefulness of combining Scrum with participatory design, since this case study only documents the combination of Scrum with a single participatory design method.

Additional studies should be performed to investigate the effect and feasibility of combining an agile development method with a participatory design method. Such studies, would be able to help developers decide if and how to adapt their existing agile development methods to better account for end-user participation. Further studies should be conducted to investigate the effect of end-user participation during the different stages of development. Such studies would be able to provide valuable insight into when the end-users are able to contribute to a project, thus be able to better pinpoint the exact stages in development where the end-users contribute. It is clear that there are different ways to facilitate end-user participation in agile development, and while the proposed way of combining Scrum and MUST worked in this setting, other approaches might be used in other settings. To get a better understanding on how the IT industry facilitates end-user participation in agile development, it would be interesting to conduct research on this. Furthermore, moving from the setting of this research into the Software industry, it would be interesting to conduct research into the employment of the adapted development method, to get a better understanding on whether the adapted development method enhances end-user participation in other settings.

Acknowledgments

We are very grateful to the 5 participants that helped us as end-users for our application. We would also like to thank Henning Nielsen who acted as Product Owner for our project. Further we would like to thank Dorthe Hinzmann, Chairman of the Local Diabetes Association, for helping us find participants for the project. Lastly we would like to thank Annebirthe Bo Hansen, Chief physician at the Department of Clinical Biochemistry, for her help in supplying knowledge about diabetes.

References

[1] Ines Anić. 2015. Participatory Design: What it is, and what makes it so great. (11 2015). https://medium.com/@i_anic/participatory-design-what-it-is-and-what-makes-it-so-great-6bad619e8735

[2] Isabelle Aujoulat, William DHoore, and Alain Deccache. 2007. Patient empowerment in theory and practice: Polysemy or cacophony? *Patient Education and Counseling* 66, 1 (4 2007), 13–20. https://doi.org/10.1016/J.PEC.2006.09.008

[3] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. 2001. *Manifesto for Agile Software Development.* Technical Report. http://agilemanifesto.org/

[4] Stephanie Chamberlain, Helen Sharp, and Neil Maiden. 2006. Towards a Framework for Integrating Agile Development and User-Centred Design. Springer, Berlin, Heidelberg, 143–153. https://doi.org/10.1007/11774129{_}15

[5] CollabNet VersionOne. 2017. *12th Annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-427987332-12th-annual-state-of-agile-report-overview/473508

[6] C Hansson, Y Dittrich, and D Randall. 2006. How to Include Users in the Development of Off-the-Shelf Software: A Case for Complementing Participatory Design with Agile Development. In *System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on*, Vol. 8. IEEE, 175c. https://doi.org/10.1109/hicss.2006.205

[7] Karlheinz Kautz. 2011. Investigating the design process: participatory design in agile software development. *Information Technology & People* 24, 3 (8 2011), 217–235. https://doi.org/10.1108/09593841111158356

[8] Finn Kensing, Jesper Simonsen, Keld Bodker, and Keld dker. 1998. MUST: A Method for Participatory Design. *Human–Computer Interaction* 13, 2 (6 1998), 167–198. https://doi.org/10.1207/s15327051hci1302{_}3

[9] Sari Kujala. 2003. User involvement: A review of the benefits and challenges. *Behaviour & Information Technology* 22, 1 (2003), 1–16. https://doi.org/10.1080/01449290301782

[10] Marta Larusdottir, Jan Gulliksen, and Åsa Cajander. 2017. A license to kill – Improving UCSD in Agile development. *Journal of Systems and Software* 123 (1 2017), 214–222. https://doi.org/10.1016/J.JSS.2016.01.024

[11] Jonathan Lazar, Jinjuan H Feng, and Harry Hochheiser. 2017. *Research methods in human-computer interaction*. http://www.worldcat.org/isbn/9780128053904

[12] José Luis Monteagudo Peña and Oscar Moreno Gil. 2007. *e-Health for Patient Empowerment in Europe*.

[13] A. Martin, R. Biddle, and J. Noble. [n. d.]. The XP Customer Role in Practice: Three Studies. In *Agile Development Conference*. IEEE, 42–54. https://doi.org/10.1109/ADEVC.2004.23

[14] Michael Muller and Allison Druin. 2002. *Participatory Design: The Third Space in HCI*.

[15] Roman Pichler. 2010. *Agile Product Management with Scrum: Creating Products That Customers Love* (1st ed.). Addison-Wesley Professional.

[16] Jenny Preece and Rombach. 1994. A taxonomy for combining software engineering and human-computer interaction measurement approaches: towards a common framework. *International Journal of Human-Computer Studies* 41, 4 (10 1994), 553–583. https://doi.org/10.1006/ijhc.1994.1073

[17] Markus Rittenbruch, Gregor McEwan, Nigel Ward, Tim Mansfield, and Dominik Bartenstein. 2002. Extreme participation - Moving extreme programming towards participatory design. In *Participatory Design Conference (PDC 2002)*, T Binder, J Gregory, and I Wagner (Eds.). CPSR, Malmoe, Sweden, 29–41. https://eprints.qut.edu.au/105421/

[18] H. Robinson and H. Sharp. 2003. XP culture: why the twelve practices both are and are not the most significant thing. In *Proceedings of the Agile Development Conference, 2003. ADC 2003.* IEEE, 12–21. https://doi.org/10.1109/ADC.2003.1231448

[19] Hugh Robinson and Helen Sharp. 2005. The Social Side of Technical Practices. In *Extreme Programming and Agile Processes in Software Engineering*, Hubert Baumeister, Michele Marchesi, and Mike Holcombe (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 100–108.

[20] Dina Salah, Richard F. Paige, and Paul Cairns. 2014. A systematic literature review for Agile development processes and user centred design integration. *ACM International Conference Proceeding Series* (2014). https://doi.org/10.1145/2601248.2601276

[21] Ken Schwaber and Jeff Sutherland. 2017. *The Scrum Guide*. http://www.scrumguides.org/scrum-guide.html

[22] Jesper Simonsen, Finn Kensing, and Keld Bødker. 2000. *Professionel IT forundersøgelse : grundlaget for bæredygtige IT-anvendelser*. Forlaget Samfundslitteratur. http://www.worldcat.org/isbn/8759308540

[23] O Sohaib and K Khan. 2010. Integrating usability engineering and agile software development: A literature review. In *2010 International Conference On Computer Design and Applications*, Vol. 2. 2–32. https://doi.org/10.1109/ICCDA.2010.5540916

[24] Hrafnhildur Sif Sverrisdottir, Helgi Thor Ingason, and Haukur Ingi Jonasson. 2014. The Role of the Product Owner in Scrum-comparison between Theory and Practices. *Procedia - Social and Behavioral Sciences* 119 (3 2014), 257–267. https://doi.org/10.1016/J.SBSPRO.2014.03.030

A longitudinal evaluation of a mobile application using two remote asynchronous usability test methods

Jakob Nymann Holgersen
Department of Computer Science
Aalborg University, Denmark
jholge14@student.aau.dk

Mathias Huse Jensen
Department of Computer Science
Aalborg University, Denmark
matjen13@student.aau.dk

Pelle Ulmer Jørgensen
Department of Computer Science
Aalborg University, Denmark
pjarge14@student.aau.dk

Abstract

Today developers, users and usability evaluators can be distributed across space and time, which makes conventional usability testing challenging, where remote usability testing can be a better suited option. This paper investigates two different remote asynchronous usability testing methods. Remote asynchronous usability testing denotes a situation where the user and evaluator attend the process at different times and different places. The methods investigated are including a diary and providing continuous updates, in addition to User-reported Critical Incident (UCI) reports when performing remote usability testing. Participants were divided into four conditions, which all received daily assignments with the ability to report usability problems in the application. The findings show that having a diary, contributes to finding usability problems in addition to also provide knowledge about how the users used the application. Furthermore, the groups who did not receive continuous updates reported the same usability problem throughout the testing period, where the groups receiving updates did not. Lastly it was shown that the users, were able to rate 59% of the identified usability problems similar to the rating given by evaluators.

Keywords Usability, Logitudinal, Remote testing, Asynchronous, End-user, Human-computer interaction

Introduction

With software development becoming more global and outsourcing of development is increasing, remote usability testing is becoming more relevant. Today developers, users and usability evaluators can be distributed across organizations, distance and time zones, meaning that conventional usability testing would be much more complex and challenging than remote usability testing [1][18]. These changes in software development imply that new methods and approaches are needed to facilitate this remote distance between the different people involved. Remote usability testing are one of these possible methods that can be an alternative to the conventional usability testing [1][2][13][14][19][22][23]. Remote asynchronous usability testing denotes a situation where the user and evaluator attend the process at different times and different places [1][4][5][20]. Remote usability testing makes testing possible both in terms of challenges with distance and differences in time. Furthermore, remote usability testing enables testing with a broader group since the testing are more convenient for the evaluators [1][22].

The approach of remote usability testing is to let the users operate the system under testing in their normal work environment [20]. The users have received training regarding identifying usability problems. Hence, when a usability problem occurs, the user should be able to report the problem. The reporting is done through a structured form, that is used to describe the problem and the steps leading up to the problem [1][4][5]. Each usability problem is sent to the evaluators to be evaluated at a late time. By doing so, the problem can be recreated by the evaluators if necessary. With remote usability testing data capturing is effective because the received problems is already assessed by severity by the users during their daily work routine. This real-time reporting prevents loss of data, as the report can be made immediately after the incident. However, this comes with a trade-off of the users natural work flow being interrupted. Furthermore, depending on the structured reports, the data are of high quality as they are easily translatable into usability problem descriptions [1][2][5].

This paper investigates two different remote asynchronous usability testing methods conducted as a longitudinal evaluation. One method is testing whether reporting usage and usability problems, through a diary, have an effect on UCI reports the users report and the quality of the feedback it is possible to gather from users. The other method is whether a continuously updated application during testing, have an effect on the UCI reports made by the users. Lastly this paper investigates whether the users are able to rate the severity of usability problems themselves.

In the following section other research on remote usability testing and user reporting are described. Following this the paper presents the remote asynchronous usability methods used in this research. Furthermore, this section also outlines the participants for this paper. Following this the findings are described. These outline the number of usability problems identified, the classification of these and the users perception of remote usability testing. Finally, these findings are discussed and advice are given regarding the practical use of this research.

Related work

Remote usability testing

Traditional user interface evaluation is conducted in a lab where users are observed directly by an evaluator, but today users can be spread across the world, which means that another approach is needed. Hartson et al. [12] outlined in 1996 a possible approach to this, being remote usability testing. Hartson et al. [12] describes different possible methods as to how the remote usability testing can be conducted, such as portable evaluator, local evaluation at a remote location, Instrumented Remote Evaluation, semi-instrumented remote evaluation and more. The semi-instrumented approach has potential for cost-effectiveness, since the user and the system gather the data [7]. This means that the evaluators only analyse the data that are relate to usability problems. However, they describe the disadvantage of being reliant on users, to identify critical incidents, with only minimal training. Of the many proposed ways to conduct remote usability testing Hartson et al. [12] presents case studies for only a few, these being remote usability evaluation using desktop video conferencing and semi-instrumented remote evaluation. Hence, they are not able to conclude if the different approaches work in practice.

Dray and Siegel [8] describes the benefits and risks of remote usability testing on a general level. The benefits that one gets are that the cost of conducting the tests is often less, there is no need for facilities to conduct the tests at and that the testing can be spread over time, since a specific time and place are not needed [20]. The risks are that there is no evaluator to help if something unexpected happens. Furthermore, it is impossible to observe indirect cues of problems, such as body language, that are not verbally expressed by the tester [9].

Self-reporting

Hartson and Castillo [11] describes a study investigating if users are able to identify and report critical incidents on their own. Furthermore, evaluators should be able to use these UCI reports to reproduce the incident later. The UCI reports can serve to collect the reactions of the users [7]. This is especially relevant when performing asynchronous remote usability testing since the users and evaluators have no direct contact and monitoring the system can be difficult [11]. The approach is to ensure that all critical incidents are reported following a template so that sufficient information is given. All users are given brief training regarding how to identify a critical incident. All users are given the same set of tasks to perform. The result was that users with little to no experience and knowledge in the field were able to identify and report critical incidents. Furthermore, the severity rating, given to the identified incidents, was similar to the rating given by the evaluators, where 83% was same severity [11]. The problem discovered are that the users often reported the critical incident at a later time, making monitoring the system at the time of the incident difficult, since data are only captured shortly before and after the report are made. It should be noted that for performing remote usability testing it was necessary to use a specific room for recording, since it was not possible to give the recording equipment to the participants, hence the participants were obligated to meet the evaluators for the test.

Möller et al. [17] investigate the frequency, accuracy and effect self-reporting have on the test subjects. The study lasted for six weeks where users should report their smart phone usage. Different settings were deployed to see their effect on the users. One group were told, at the beginning of the test period, that they should remember to report their usage. The second group got periodical reminders telling them to remember to report, where the last group were reminded every time they closed an application. Möller et al. [17] showed that the users made reports ranging from 70% to 40% of their actual usage. Furthermore it was shown that users often overestimated their usage[6][10][17]. Aspects such as forgetfulness or intentional miss-reporting can be a factor [15]. Furthermore, the periodical reminders resulted in most reports. The users said that being reminder to often was uncomfortable and burdensome. It should be noted that most of the participants in this study were students which can have an effect on their usage of their phone during the day compared to the participants that were working.

Steves et al. [21] describes two case studies, the first in which collaborators the tool for evaluation over several months. The data was collected through self-reporting by the users through diaries, where after survey questionnaire and interviews was conducted. The study was conducted in a real work environment to get an understanding of users in their own environments and by that users would provide clues about critical aspects of the real work situation [24]. The second case study was conducted as a usability inspection in

a laboratory. In this study independent evaluators assessed the tool by completing different use scenarios and examine how well it fitted inspection criteria. They found that the approach of supplementing auto logging with a diary where participants provided qualitative information on the problems identified was beneficial.

Comparison of remote usability testing methods

Bruun et al. [3] systematically compares three methods for remote asynchronous usability testing: UCI, forum-based online reporting and discussion, and diary-based longitudinal user reporting. To prepare the user for remote testing, they train them in how to identify and categorize usability problems. This is done using a minimalist approach, that is strictly remote and asynchronous, where the users receive written instructions via email. They show that the remote asynchronous usability testing identifies about half the usability problems compared to the conventional method, however at a much less time effort. Furthermore, the severity rating made by the participants, 77% for UCI, 66% for forum and 45% for diary, is rated the same categorization as by the evaluators. With these findings they deem remote asynchronous methods an appealing possibility for usability testing in many software projects. It should be mentioned that the system used for this research was a finished released product, where often usability testing is conducted on versions of a product that are still in development. In addition, Bruun et al. [3] show that for critical and serious problems, the Diary collection method identified around 50% of the problems identified in a lab settings, however it was less time-consuming with only 30% of the total hours spent compared to lab. Following this they suggested that this time could be further reduced if the diary were combined with the problem format that was used with UCI, hence the analysis of the diaries would be shorter. Lastly, Bruun et al. [3] experienced that 38% of the usability problems identified through the diaries could not be categorized properly.

Research Method

We have conducted an empirical study of longitudinal remote asynchronous usability testing. The methods investigated are including a diary and providing continuous updates, in addition to User-reported Critical Incident (UCI).

Participants: A total of 16 users, 10 female and 6 male, participated as usability testers divided into four conditions. Table 1 shows the demographics of the users, with 10 type 1 and 6 type 2 diabetes. These demographics have been compiled using data from a preliminary questionnaire send to the users, asking for contacts and basic information. The users were recruited through the local diabetes association's news email and diabetes associated Facebook groups. The authors acted as evaluators when analyzing the usability problem instances, this included rating and generating the problem list of usability problems.

Setting: The users could carry out the tasks where and whenever they wanted. They were not forced to complete the daily tasks at a specific time or in one session.

Materials: The application used for testing was an application we created for assisting diabetics' daily life. To facilitate longitudinal remote testing four different versions of the application were released on four different test branches on the Google Play Store that the users had access to. Each version corresponding to a specific group, thus it was possible to distinguish between the applications the users had. Through the testing period the applications for the groups that should receive updates were updated three times. The first update was after 4 days of testing, the second after 8 days and the last update was after 12 days. All updates were based on fixing the reported usability problems.

Procedure: We have conducted longitudinal remote asynchronous usability tests to identifying possible usability problems within a selected mobile application. Conditions were changed between the different usability tests, to investigate if the changed aspects influenced the usability problems reported. All condition had the ability to report using UCI reports, in the application itself.

The different conditions of the longitudinal remote asynchronous usability tests, that were changed depending on the user groups were; if the group received continuous updates to the application they were testing and if they had the possibility of writing feedback in the form of a diary. Table 1 shows the condition for each group.

The testing period for each user was 14 days. At the beginning of the test period each user received training in form of an instruction guide, explaining what a usability problem is and how to report it. This included examples of what a usability problem could look like. In addition, step-by-step instructions on how to download and install the application were provided.

To encourage the users to continuously use the application, push-notifications were triggered from the application three times each day, at 8 am, 1 pm and 8 pm. This was done to remind the users to use the application and report any problems experienced. Furthermore, 2 tasks were sent to the users every day, at 1 pm, via SMS and email to ensure that the users had a use for the application each day, in addition to any regular usage. All tasks sent to the users were performed by the evaluators beforehand ensuring that the tasks were

	Update			**No Updates**		
Diary	**Group 1**	Type	Treatment	**Group 3**	Type	Treatment
	Female (57)	Type 2	Pen	Female (33)	Type 1	Pen
	Female (57)	Type 1	Insulin	Female (38)	Type 1	Pen
	Male (49)	Type 2	Tablets	Female (62)	Type 2	Insulin
	Male (59)	Type 1	Pump	Male (69)	Type 1	Pump
No Diary	**Group 2**	Type	Treatment	**Group 4**	Type	Treatment
	Female (25)	Type 1	Insulin	Female (22)	Type 1	Pen
	Female (61)	Type 2	Medicine	Female (35)	Type 1	Pen
	Male (41)	Type 1	Pen	Female (67)	Type 1	Pen
	Male (69)	Type 2	Tablets	Male (61)	Type 2	Pen

Table 1. Users' demographics and group distribution

solvable. The testing by the users were asynchronous meaning that they could use the application and report identified usability problems at any time.

Data collection: Data were collected with UCI reports and diary entries from the users. The UCI reports required a description of the actions leading to the problem which occurred. Furthermore, a severity rating were given to the problem, describing how the problem affect their use of the application. Furthermore the user described if it was possible to complete the task. The diaries contained a description of what the user used the application for that day. Additionally, the diary also had a description of a few good and bad experiences with the application. Interviews were conducted following the end of the testing period. In the interviews the users were asked about their reported problems and what they felt about the remote usability tests.

Data analysis: The UCI reports were translated into usability problems based on: the description of the UCI report, the specified page where the problem occurred and the element on the page that triggered the problem. The UCI reports were collected, and the evaluators assigned a severity rating to the identified usability problems. This severity rating was based on four criteria: a description of the problem, problem type (application or usability), the page on which the problem occurred and what element on the page caused the problem. This was done in collaboration, where each evaluator voted for the severity rating of the problem. If any disagreements occurred the problem was discussed, to reach an agreement of the severity rating. The rating was assigned independently from the rating given to the problem by the users that reported the problem. The usability rating given was either critical, serious or cosmetic based on the severity of the reported problem [16].

The diary entries were transformed into usability problem instances by first identifying the diary entries describing problems. Each of these diary entries were then read by each evaluator from which the evaluators then created a problem description based on the diary entry. If there were any disagreements regarding the wording or meaning of the problem description, it was discussed, until an agreement could be reached. Furthermore, from the diary entry the relevant page and elements involved with the problem were also identified.

The different reported usability problems were then assigned to the appropriate groups that reported them. This was done to gain an overview of who reported the problems and how many they each reported. The follow-up interviews were analyzed by the three evaluators together by going through all recorded interviews and highlighting important and relevant statements made by the users. The answers were collected as qualitative data, to either support or clarify the results from the UCI reports and diaries.

Findings

UCI reports and diary entries

A total of 65 UCI reports were received from 6 out of the 16 users, 2 reports were discarded as they did not describe a usability problem. No UCI reports were received from group 1, where every report in group 3 were contributed by a single person. In group 2 two of the users submitted UCI reports, where in group 4 three persons contributed with UCI reports. Person 4 in group 4 initially sent UCI reports as email instead of using the report function of the application. Table 2 shows the distribution of the amount of UCI reports for each group and who submitted these.

	Updates	No updates
Diary	**Group 1**	**Group 3**
	Person 1: 0	Person 1: 0
	Person 2: 0	Person 2: 0
	Person 3: 0	Person 3: 0
	Person 4: 0	Person 4: 24
Total reports	0	24
No diary	**Group 2**	**Group 4**
	Person 1: 0	Person 1: 0
	Person 2: 0	Person 2: 19
	Person 3: 6	Person 3: 2
	Person 4: 4	Person 4: 8
Total reports	10	29

Table 2. Number of UCI reports from each person and the respective groups

Figure 1 shows the frequency of which days the UCI reports were received. It can be seen that the number of daily reports were declining throughout the testing period except on the last day of testing where there were an unexpected increase in the number of UCI reports received.

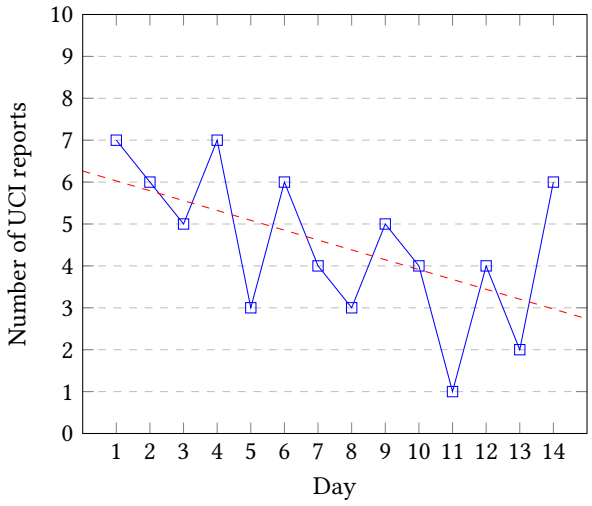

Figure 1. Frequency of when UCI reports received - Red line is the trend line and the blue is the UCI reports

Throughout the test period we received a total of 16 diary entries during the 14 day test period. 2 of the entries were from a single person in group 1, where 14 were from 3 different persons in group 3. 11 of these were from Person 1 with the remaining from person 2 and 3. Of these 16 diary entries, 4 were discarded, as they did not describe any problem, but rather instead positive feedback. Of the remaining 12 diaries, 1 diary mentioned two instances of usability problems, while the rest described one each. Table 3 show an overview of the diary entries received. Through analysis of the diary entries, 13 instances of usability problems were identified. The shortest diary entry was 63 characters long and the longest being 504 characters. The average length of a diary was 238 characters, which can be contributed to writing on a smartphone.

	Updates	No updates
Diary	**Group 1**	**Group 3**
	Person 1: 0	Person 1: 0
	Person 2: 2	Person 2: 2
	Person 3: 0	Person 3: 1
	Person 4: 0	Person 4: 11
Total reports	0	24

Table 3. Number of diary entries from each person and the respective groups

Together with the diary entries a total of 76 instances of usability problems was identified and a severity rating was given, with the following distribution: 25 Critical, 19 Serious and 32 Cosmetic. Table 4 gives an overview of the distribution of these usability problem instances and the usability problems. A total of 45 usability problems were identified through the combination of both UCI reports and diary entries from the total of 76 instances of problems.

	UCI (63)	Dairy (13)	Total usability problems
Critical (25)	14	1	14
Serious (19)	7	2	7
Cosmetic (32)	22	4	24
Total (76)	43	7	45

Table 4. Total amount of identified usability problem instances, in parenthesis and the number of usability problems, both for UCI and Dairy alone and total combined, rated by the evaluators

In Table 5 a few of the identified usability problems can be seen. These usability problems were created based on the submitted UCI reports and diary entries.

Description	Page	Severity
Missing feedback leading to confusion when saving an insulin calculation	Insulin Calculator	Critical
Unable to find and change the date in order to display measurements in a given range.	Graph	Critical
Unable to sort the food list	Meal Plan	Serious
Confusion between decimal separator, decimal point or decimal comma	Insulin Calculator	Cosmetic
...	...	...

Table 5. Snippet of identified usability problems

Comparison with or without the condition

To outline how the different groups contributed, we compared the results from the groups which made diary entries to the groups which did not and the groups who received updates to the groups that did not.

Updates vs No Updates
It can be seen that updates have had an effect on the UCI reports made by the users, when looking at the groups that did receive updates compared to the groups that did not. The update groups submitted 10 UCI reports, all describing usability problems, while the no update group submitted 53 UCI reports describing instances of usability problems, where 36 of these were different usability problems. There was an overlap in the reports submitted by the groups that received updates and those that did not.

Some of the reported usability problems were fixed in the continuous updates to the application. To further investigate the effect of having updates, follow-up interviews were held with the users at the end of the test period. One user from group 2 mentioned that she: "I experienced that the updates corrected the problems i reported. This was motivating to continue testing as I wanted to report problems when I know they are being fixed." Another user from group 3, which did not receive updates, expressed frustration in three diary entries and continuously reported in 7 UCI reports the same usability problem, which had been fixed in the first of the three updates. An example from the diary entry: "Frustrated with your tasks, many of them can not be completed, may soon expect a feedback on whether it is trick questions or if your app does not work as intended. Seemed to be wasting time, I would be happy if the app worked, but apparently it doesn't. You are welcome to call or send the instruction manual that has been asked for." The user felt that he was wasting his time and did not feel like that the task was completable, even though all tasks were tested beforehand.

Diary vs. No Diary
The diaries contributed to the identified UCI problems. The total number of usability problems can be seen in Table 4. The diaries contributed with 7 instances of usability problems, where 5 of these was mentioned in the UCI reports, which means 2 additional usability problems were identified with the use of diaries.

Besides 2 usability problems, the diaries contributed with knowledge about how and which parts of the application the users used. The diaries showed that the Meal planner functionality and the Insulin calculator functionality were frequently mentioned. An example from one diary entry was: "I really like that you can calculate via the app how many carbohydrates are in the entire meal where you usually have to stand and calculate it together on either a calculator or in the head before you enter total carbohydrates for the entire meal". Furthermore, the users were able to provide suggestions to additional functionality through the diary. An example from one of the diary entries was: "It is simple to know what to press and how it works but I could use the ability to add custom food items that are not in the list in the Meal planner".

Rating of usability problems

To investigate if the users were able to rate usability problems themselves, we compared the instances of usability problems rated by the users and by the evaluators. Table 6 gives an overview of how many instances of usability problems were given the same or different ratings by the users and evaluators. The users rated the instances of usability problems as either cosmetic or critical. Hence only 3 of the 63 instances of usability problems (4.76%) identified from the UCI reports were categorized as serious.

Looking at the ratings the users made, 26 of the instances of problems rated by the users had been rated different by the evaluators, meaning that 41% of the instances of usability problems were rated differently. For example, a critical problem reported by a user was: "Unable to enter 0 carbohydrates in the insulin calculator", where the problem have been rated cosmetic by the evaluators. Another example was a problem that was rated cosmetic by the user and critical by the evaluators, this problem was: "Application is stuck in landscape mode after using the graph page". This showed that despite the users receiving instructions as to how to rate the severity of usability problems, usability experts should not rely only on the rating given by the users.

	Update				**No Updates**			
Diary	**Group 1**	Common	Users	Evaluators	**Group 3**	Common	Users	Evaluators
	Critical:	-	-	-	Critical:	13	11	0
	Serious:	-	-	-	Serious:	0	0	4
	Cosmetic:	-	-	-	Cosmetic:	0	0	7
No Diary	**Group 2**	Common	Users	Evaluators	**Group 4**	Common	Users	Evaluators
	Critical:	1	3	1	Critical:	5	6	2
	Serious:	0	1	1	Serious:	1	1	7
	Cosmetic:	4	1	3	Cosmetic:	13	3	1

Table 6. Overview of severity rating of usability problems instances by users and evaluators

Lack of responses

To gain knowledge, about the users' perception of the asynchronous remote usability test, follow-up interviews were conducted. In Table 2 the number of UCI reports reported by each user over the test period of 14 days are shown. A total of 63 UCI reports were received from the 16 users, with a distribution of top 4 users accounting for 90% of the reports and 10 of the 16 users not making a single UCI report.

To investigate the lack of and distribution of UCI reports all users were contacted for a follow-up interview. In Group 1, two of the four users did not respond despite multiple attempts at contacting them over a period of two weeks. In the beginning of the testing period they received extensive guidance on how to install the application over the phone by us. The remaining two users said that they had ended the test period after two days, meaning that they had continued to receive mails, SMS and push-notifications every day for the remaining 12 days, without contacting us.

As the understanding of the UCI page could have had an effect on the lack of reports, the users were asked if the UCI reporting page was confusing and hard to use. 10 of the 16 users said that they thought that the UCI reporting page looked easy to use, despite of this 5 of these users did not create any UCI reports.

In Group 3, Person 2 mentioned that she did not report because she thought that she was the problem not the application. In addition to this Person 4 said that she: "Found problems I did not report because I thought they were minimal". These statements contradict the instructions that each user received at the start of the testing period, describing what a usability problem is, with examples, and encouragement to report everything the users encountered. Furthermore, Person 4 from Group 4 said that he: "Did not use the reporting in the app. I Didn't read the guide explaining how to report."

Several of the users mentioned that their daily life got in the way of using the application. Person 4 from Group 3 said that she "Was sick so only used the app some of the days" while Person 4 from Group 4 said that he "was busy at work, so did not use the app every day". However, by having this study as a longitudinal study, the users were still able to participate in the remote usability testing since they had a longer period of time to participate than if lab testing had been used.

Despite the lack of UCI reports, 10 of the 16 users contacted in the follow-up interviews mentioned that the daily push-notifications, SMS and mails with small assignments reminded them to use the application. The push-notification at the end of the day specifically reminded the users to create UCI reports and diary entries regarding their experiences of that day, despite that, only 6 users contributed with UCI reports.

Discussion

This section discusses interesting observations and findings. Furthermore, these observations and findings are discussed in relation to related work. This is followed by a discussion of the practical relevance of performing remote usability testing with regards to providing updates during the test period and the impact of supplementing UCI reports with diary.

Some of the most surprising observations and findings from this study were the results from the remote asynchronous usability test, that showed that few users actively participated by creating UCI reports and diary entries. This was surprising given the amount of time and effort put into ensuring that the users understood the tests and continuously reminded them to use the application. These reminders consisted of push notification at 8 am, 1 pm and 8 pm, this also included the daily tasks send at 1 pm by SMS and email for 14 days. Besides reminders, the users received an 8 pages instruction guide with pictures showing them step-by-step how they should install the application, examples of what a usability problem could be and contact information they could use if they encountered any problems or needed help. Despite this the follow-up interviews and analysis of the UCI reports and diaries showed that that 2 users from Group 1 had ended

the test period after two days, without making contact. One of the reasons given was: "I had to end the testing as i did not have the time to participate". Despite multiple attempts, we were not able to get in contact with the other user that ended the test period prematurely.

The examples of the usability problems were provided, to ensure that the users had an understanding of what a usability problem could look like. These were based on examples used by Bruun et al. [3], who used these to successfully train their users. Despite this, the definition of a usability problem, was not clear for all users, which became apparent through the follow-up interviews. Several users mention that they did not report every problem they encounter, where one participant stated that they did not see them as usability problems but thought the problem was caused by themselves and not related to the usability of the application. This is an interesting observation as all users received an instruction guide specifically explaining, using examples of what a usability problem is and how they could identify them. However, results indicate that the users did not read these instructions as they did not report the problems they encountered, which was further supported by a user who mentioned that they did not read the instruction guide.

Given the nature of remote asynchronous usability testing, we observed a decreasing lack of commitment. This can be seen in the number of UCI reports, were 10 users submitted 0 reports through the test period. Furthermore, this being a longitudinal evaluation had a further negative effect on the response rate. An example of this is that in the first week, responses were received from the 6 participating users, where in the second week of testing only 2 users continued creating reports. Furthermore, we have observed a tendency through the longitudinal evaluation, showed that in our case we received fewer reports as the testing progressed. The users were reminded during the two-week test period to use the application by SMS, email and push-notifications. However, this was one-way communication, which might have had an effect on the users' commitment, since the users could ignore this as they never had to reply.

In relation to related work, we presented results where usability problems were rated by both the users and the evaluators. Looking at the literature with regards to users severity rating of usability problems, Bruun et al. [3] and Castillo et al. [5] reported a similarity of ratings at 77% and 83% respectively, where the similarity in this paper is lower at 59%. This can be contributed to the way users were trained, as the training was done by the users reading the sent instructions specifying what a usability problem is and the different severity of these. Hence, if the users did not read the instructions, they had no training in identifying and categorizing usability problems. An example was that Person 4 from Group 4 said that he: "Did not use the reporting in the app. I Didn't read the guide explaining how to report". Given the similarity rating at 59% it can be seen that we cannot rely solely on the severity rating given by the users. This follows Hartson et al. [12], which describes solely relying on the users to rate and identify usability problems, with minimal training, as a disadvantage of remote evaluation.

Möller [17] showed that receiving periodical notifications had a positive effect, when you needed to engage users in reporting and using the tested application. For this research periodical notifications were used. In the follow-up interviews the users were asked regarding these notifications and whether these had helped or had been an annoyance. Multiple users mentioned that notifications did work as a daily reminder to use the application. One user said: "Notifications help remind me to use the app. Some days when I was busy, I had forgotten about the app but the notifications reminded me about it". Despite this, the amount of UCI reports and diary entries showed that not all users reported usability problems.

Dray and Siegel [8] and Hartson et al. [12] described different approaches to remote usability testing and the challenges of these. For this research we encountered some of these problems, especially regarding the problems of the evaluators not being able to observe and help with unexpected encounters. In general for this research being able to monitor the users better would have been beneficial, since it would have been possible to identify the causes behind the lack of reporting from some users. Two different approaches to account for these problems can be taken. In terms of direct monitoring the evaluation will not be asynchronous, as the separation of time when testing and the monitoring will have to been synchronised. This would require an active participation by an evaluator or a test monitor, hence in line with synchronous remote testing. In terms of indirect monitoring, hence keeping the test asynchronous, Hartson and Castillo [11] experimented with indirect monitoring of a users system. The challenge discovered was that capturing the moment a problem occurred was difficult, since there was no indication of when a problem would occur. In addition to this, they experimented with monitoring the system shortly before and after users created UCI reports, but found that these reports were often not created immediately following an occurred problem.

Besides having each user fill out UCI reports doing the testing period, diary data collection was added to 8 the users. This was done to gain qualitative data from the users about the daily usage of the application, as found by Steves et al. [21] this provide a better understanding of the UCI reports made by the users that were supplemented by a diary. The diary data collection was structured as traditional diary with two

additions for the users to explicitly express their best and worst experiences in the application on that day. This lead to data being able to be easier to rate compared to Bruun et al. [3], where 38% of their identified usability problems were not able to be rated.

In regard to practice, this paper presented different methods to conduct remote asynchronous usability testing and their respective results. The lessons learned through this research will be used to give advice regarding the practical implications of performing remote usability testing. The first thing learned was that having a structured method of reporting was important for handling the data the user generated. Having the user fill a UCI form made translating these into usability problems fast and efficient. Having a diary enabled the users to better express their thoughts about the application, but in this research the diary entries were often used to explain already reported UCI problems and not describing the general usage of the application. Having a diary was seen as beneficial since it still enabled additional insight into the users thoughts about the application.

Another consideration is how the users are monitored during testing. For this research there were minimal monitoring of the users as it was expected that the users could and would describe what happened when a problem occurred. Since the structure of the UCI form, promoted a detailed template for reporting when and where a problem occurred. In addition, if the application was monitored in a way that made it possible to collect data about the page the users were using when a problem occurred. We would then not rely solely on the users, since we would be able to verify the reported problems and gain additional data regarding application usage. The difficulty here would be to ensure that data were collected when a problem occurred and not when the user reported it. As mentioned by Hartson and Castillo [11] users will not always report a problem when it occurs but might first report it at a later time.

Conclusion

The goal of this paper was to investigate different methods to remote asynchronous usability testing. The paper presented methods using a diary and methods using continuous updates. The findings showed, that having a diary, contributes to identifying usability problems in addition to also provide knowledge about how the users used the application. Furthermore, the groups who did not receive continuous updates reported the same usability problem, where the groups receiving updates did not. In addition to this, it was shown that the users, were able to rate 59% of the identified usability problems similar to the rating given by evaluators.

We acknowledge the limitations of our work, and especially regarding the evaluation of the different methods to remote asynchronous usability testing. We acknowledge that the limited number of UCI reports, results in few users having a high effect on the perceived effectiveness of the methods. We acknowledge that for this study the perceived effectiveness of the different methods was dependent on the users in these.

Additional studies should be performed investigating the effect of giving users continuous updates throughout the testing period. For this research the group given updates produced fewer UCI reports than the other groups and the users mentioned that they could see that their reported problems were fixed which worked as a motivation to continue participating in the testing. Additional testing should be done with a larger group of users and over a longer period of time to see the full effect of updates in addition to UCI reports. Furthermore, research should be conducted regarding the usage of a diary when doing remote asynchronous usability testing, as the diary can be used in different ways by the users, thus further research regarding these possible different uses of a diary is appropriate.

Acknowledgments

We are very grateful to the 16 users that helped us by participating in the remote usability testing of our application. We would like to thank Dorthe Hinzmann, Chairman of the Local Diabetes Association, for helping us find participants for the project.

References

[1] Morten Sieker Andreasen, Henrik Villemann Nielsen, Simon Ormholt Schrøder, and Jan Stage. 2007. What Happened to Remote Usability Testing?: An Empirical Study of Three Methods. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 1405–1414. https://doi.org/10.1145/1240624.1240838

[2] A J Bernheim Brush, Morgan Ames, and Janet Davis. 2004. A Comparison of Synchronous Remote and Local Usability Studies for an Expert Interface. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems (CHI EA '04)*. ACM, New York, NY, USA, 1179–1182. https://doi.org/10.1145/985921.986018

[3] Anders Bruun, Peter Gull, Lene Hofmeister, and Jan Stage. 2009. Let your users do the testing: A comparison of three remote asynchronous usability testing methods. In *Conference on Human Factors in Computing Systems - Proceedings*. 1619–1628. https://doi.org/10.1145/1518701.1518948

[4] Jose Carlos Castillo. 1998. *The User-Reported Critical Incident Method for Remote Usability Evaluation*. Ph.D. Dissertation. Virginia Tech. https://vtechworks.lib.vt.edu/handle/10919/35513

[5] José C Castillo, H Rex Hartson, and Deborah Hix. 1998. Remote Usability Evaluation: Can Users Report Their Own Critical Incidents?. In *CHI 98 Conference Summary on Human Factors in Computing Systems (CHI '98)*. ACM, New York, NY, USA, 253–254. https://doi.org/10.1145/

286498.286736

[6] Frank P Deane, John Podd, and Ron D Henderson. 1998. Relationship between self-report and log data estimates of information system usage. *Computers in Human Behavior* 14, 4 (1998), 621–636. https://doi.org/10.1016/S0747-5632(98)00027-2

[7] Elisa M del Galdo, Robert C Williges, Beverly H Williges, and Dennis R Wixon. 1986. An Evaluation of Critical Incidents for Software Documentation Design. *Proceedings of the Human Factors Society Annual Meeting* 30, 1 (1986), 19–23. https://doi.org/10.1177/154193128603000105

[8] Susan Dray and David Siegel. 2004. Remote Possibilities?: International Usability Testing at a Distance. *Interactions* 11, 2 (2004), 10–17. https://doi.org/10.1145/971258.971264

[9] Jon Froehlich, Mike Y Chen, Sunny Consolvo, Beverly Harrison, and James A Landay. 2007. MyExperience: A System for in Situ Tracing and Capturing of User Feedback on Mobile Phones. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys '07)*. ACM, New York, NY, USA, 57–70. https://doi.org/10.1145/1247660.1247670

[10] Craig Hartley, Mark Brecht, Peter Pagerey, Gerald Weeks, Alphonse Chapanis, and Douglas Hoecker. 1977. Subjective time estimates of work tasks by office workers. *Journal of Occupational Psychology* 50, 1 (1977), 23–36. https://doi.org/10.1111/j.2044-8325.1977.tb00355.x

[11] H Rex Hartson and José C Castillo. 1998. Remote Evaluation for Post-deployment Usability Improvement. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '98)*. ACM, New York, NY, USA, 22–29. https://doi.org/10.1145/948496.948499

[12] H Rex Hartson, José C Castillo, John Kelso, and Wayne C Neale. 1996. Remote Evaluation: The Network As an Extension of the Usability Laboratory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '96)*. ACM, New York, NY, USA, 228–235. https://doi.org/10.1145/238386.238511

[13] Melody Y Ivory and Marti A Hearst. 2001. The State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Comput. Surv.* 33, 4 (12 2001), 470–516. https://doi.org/10.1145/503112.503114

[14] F S H Krauss. 2003. Methodology for remote usability activities: A case study. *IBM Systems Journal* 42, 4 (2003), 582–593. https://doi.org/10.1147/sj.424.0582

[15] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. 2006. A Practical Approach to Recognizing Physical Activities. In *Pervasive Computing*, Kenneth P Fishkin, Bernt Schiele, Paddy Nixon, and Aaron Quigley (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–16.

[16] Rolf Molich. 2000. *Usable Web Design (In Danish)*.

[17] Andreas Möller, Matthias Kranz, Barbara Schmid, Luis Roalter, and Stefan Diewald. 2013. Investigating Self-reporting Behavior in Long-term Studies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2931–2940. https://doi.org/10.1145/2470654.2481406

[18] John Murphy, Steve Howard, Jesper Kjeldskov, and Steve Goschnick. 2004. Location, location, location: Challenges of Outsourced Usability Evaluation. (2004).

[19] Helen Petrie, Fraser Hamilton, Neil King, and Pete Pavan. 2006. Remote Usability Evaluations With Disabled People. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 1133–1141. https://doi.org/10.1145/1124772.1124942

[20] J Scholtz. 2001. Adaptation of traditional usability testing methods for remote testing. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*. 8 pp.–. https://doi.org/10.1109/HICSS.2001.926546

[21] Michelle Potts Steves, Emile Morse, Carl Gutwin, and Saul Greenberg. 2001. A comparison of usage evaluation and inspection methods for assessing groupware usability. In *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work - GROUP '01*. ACM Press, New York, New York, USA, 125. https://doi.org/10.1145/500286.500306

[22] Thomas Tullis, Stan Fleischman, Michelle Mcnulty, Carrie Cianchette, and Marguerite Bergel. 2002. An empirical comparison of lab and remote usability testing of Web sites. (2002).

[23] Ryan West and Katherine Lehman. 2006. Automated Summative Usability Studies: An Empirical Evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 631–639. https://doi.org/10.1145/1124772.1124867

[24] Dennis Wixon and Chauncey Wilson. 1997. Chapter 27 - The Usability Engineering Framework for Product Design and Evaluation. In *Handbook of Human-Computer Interaction (Second Edition)* (second edi ed.), Marting G Helander, Thomas K Landauer, and Prasad V Prabhu (Eds.). North-Holland, Amsterdam, 653–688. https://doi.org/10.1016/B978-044481862-1.50093-5

Customer and end-user participation in software development: A survey of the Danish software industry

Jakob Nymann Holgersen
Department of Computer Science
Aalborg University, Denmark
jholge14@student.aau.dk

Mathias Huse Jensen
Department of Computer Science
Aalborg University, Denmark
matjen13@student.aau.dk

Pelle Ulmer Jørgensen
Department of Computer Science
Aalborg University, Denmark
pjarge14@student.aau.dk

Abstract

Customer and end-user participation are essential for the successful development of products and services, particularly when developing new products. With the shift of development process paradigm, changing from the traditional waterfall model towards the agile manifesto, the customer and end-users are more in focus. The agile manifesto however, does not specify any methods or techniques to facilitate end-user and customer participation during the development. This paper investigates the Danish software industry regarding the practical application of customer and end-user participation in modern development methods. This includes when and which techniques and methods are used, in addition to identifying possible obstacles. This is done through an online questionnaire send to 67 Danish software companies of varying size, product type and employed development methods. This paper shows that Scrum was a dominating development method with 79.2% of respondents using it. Furthermore, 51.5% of respondents thought that there were insufficient end-user participation in their current development process. The most used techniques for facilitating participation were Interview, Demonstration, User stories and Qualitative usability testing. The primary identified obstacles for additional customer and end-user participation were Lack of time, Geographical challenges and Lack of staff.

Keywords Human-Computer Interaction, System development, End-user, Customer, Survey, Questionnaire, Agile development, Development method

Introduction

In 2001 the agile manifesto described a different approach as to how software development could move away from the traditional development methods inspired by the waterfall method, to become more agile [1][4][33][36]. Agile development methods have since grown to be the most used approach as seen in a report from 2017, which shows that 97% of the 1492 respondents uses a form of an agile development method. When asked why they have chosen to work agile, 71% answered that the ability to change priorities is the most important aspect [18]. This ability to rapid changing priorities are important when developing software to customers and end-users, as these groups often change their mind on the requirements for the product. Hence, this ability to change is a core cornerstone of the agile development methods. While several methods for facilitating customer and end-user participation in software development exist, the agile manifesto does not propose a specific method for facilitating customer and end-user participation in the development, nor when this participation of the customer and end-user is most beneficial [3][37].

Customer and end-user participation is considered essential for successful product and service development, especially regarding development of new products [5][6]. This is especially applicable when cost is related to changing technologies, as customer and end-user feedback and collaboration is key to assessing technology feasibility and reducing risk [29][34]. It would be interesting to examine how companies in the software industry facilitates customer and end-user participation in their development.

This paper presents a survey of the Danish software industry with a focus on how, when and which methods are used to facilitate customer and end-user participation in development. This is done as an exploratory study to gain a picture of how Danish software companies are handling customer and end-user participation, and by that how they alter their development method to ensure this. A customer is defined as the company / person receiving the developed service / software, while the end-user is defined as the user, who ultimately uses the product.

In the following section we describe previous research on how software companies ensure usability of their product. Furthermore, research on conducting surveys and ensuring responses are also described. Following this the research method are outlined describing the procedure followed to

construct the survey and to distribute the survey. Then findings are presented outlining the answers given by the respondents. Finally, the findings are discussed by highlighting unexpected results and discussing the practical reality of this research.

Related work

Surveys investigating usability of software in the industry

Bak et al. [2] conducted in 2008 a combined questionnaire survey and interview study to highlight obstacles for deploying usability evaluations in software development organizations. To distribute the questionnaire 74 organizations were contacted by telephone and the questionnaire was sent to the representatives they came in contact with. After 2-3 weeks, they sent out email reminders and further two weeks after, they contacted the organizations again by telephone. In the questionnaire Bak et al. [2] wants to investigate the respondents understanding of usability before finding obstacles. Here they differentiate between customer and user, which is important as the user (end-user) might not be the customer of the product. 39 of the 74 contacted software development organizations responded and results show that the most prominent obstacles were perceived resource demands and the mindset of developers. The obstacle of a developers mindset was found to be more than just programming related. The mindset was more an ignorance of usability engineers input and a resistance towards changes to the product [2][30][32]. It should be noted that all these organizations are located within a limited geographical area.

Gulliksen et al. [31] conducted in 2003 a survey of usability professionals in Sweden. The survey identified background, experiences, the type of employment, organization and products/systems, the software development process being used and some key success factors for usability work. Furthermore, Gulliksen et al. [31] investigated, which stages of the development process facilitated user participation, giving an idea of how this has an impact on the usability. A total of 194 responded. The results show that management support and project management support are essential for the usability worker. Furthermore, they show that usability and user participation have low priority in the projects. It should be noted that the research is conducted in 2003, only a few years after the official agile manifesto from 2001, meaning that the survey is highly affected by respondents using non-agile development methods. In addition, Gulliksen et al. [31] do not differentiate between costumer and end-user, hence referring to both as "user" which can have had an effect on the responds as each respondent might interpret what a user is differently.

Enhancing response rate of web surveys

Fan and Yan [28] performed a systematic review investigating what factors have an effect on the response rate of web-surveys. They outline four steps that influence the response rate of a web-survey. These steps are survey development, survey delivery, survey completion and survey return. Regarding survey development the topic of the survey needs to have a high salience (relevance to the respondent) [25][28]. Furthermore, how long the survey takes to answer will affect the response rate as surveys generally have a negative linear correlation between length and responses [20][28]. Regarding survey delivery there are different forms, these being: pre-notification, email invitations and email reminders. When delivering the survey via email the challenges faced are the increased use of spam filters and the possibility of the survey being buried beneath other emails and by that not being seen by the respondent [21][22][28]. Regarding survey completion Fan and Yan [28] describes different groups of factors that can have an effect, these being: society-related factors, respondent-related factors, and design-related factors. The society-related factors are factors such as internet resources, computer-literacy, survey fatigue and the public's attitude towards the survey industry [23][27][28]. Regarding respondent-related factors an example is that for the respondent the cost of answering the survey needs to be outweighed by the incentive of answering the survey [26][28]. For example, keeping questionnaires short and easy can reduce perceived costs for respondents. The last step is survey return which is regarding the survey software used to ensure easy return of answers and data safety. While Fan and Yan [28] present the different aspects of conducting a survey they do not themselves conduct a survey documenting the effectiveness of their own recommendations.

CollabNet VersionOne [19] describes them-self as an enterprise Value Stream Management leader that accelerates high value software. Each year, starting from 2006, CollabNet produces a report called Annual State of Agile development, where companies worldwide have been surveyed to investigate which and how agile methods and techniques are used by the industry. In Figure 1 results from the 12 reports, spanning from 2006-2017, on agile development methods used is shown. These show that Scrum is the most prevalent development methods used. Furthermore, it also shows the the popularity of Scrum related methods are rising. The Other option includes: DSDM, AgileUp, Feature Driven Development, Lean, OpenUp, Agile Modeling, Crystal, Kanban, Iterative Development and the Spotify Model. It should be noted that the sizes of the companies surveyed, ranged from less than a 1000 employees to more than 20.000, where for example in the 2017 report 39% of answers were from companies with less than 1000 employees.

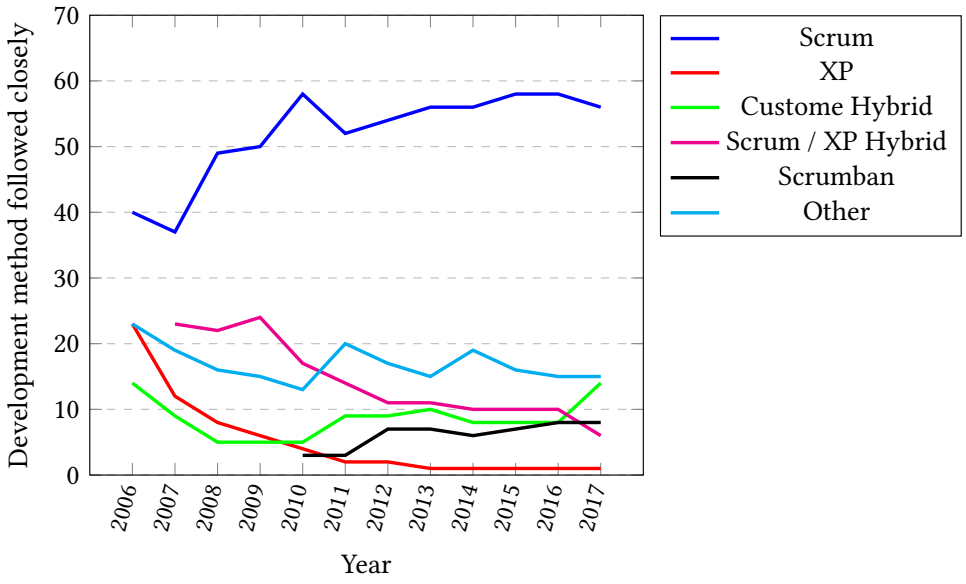

Figure 1. The state of agile development methods from 2006 to 2017 [7][8][9][10][11][12][13][14][15][16][17][18]

Research Method

Procedure: We have conducted a survey by means of an online questionnaire. This questionnaire was based on 5 interviews with participants from 3 difference software companies, followed by a pilot questionnaire spanning one week, where we received feedback from 2 participants. The questionnaire was open to anyone with the link, and recipients in the software companies were asked to forward the questionnaire within the companies. Social media was also utilized, where posts were made asking for respondents relevant for this survey. This included posts in closed Facebook groups and posts on the authors personal feeds, both on Facebook and LinkedIn. The questionnaire was also posted on forums deemed relevant to our research, this included: Computerworld, Amino, Nettips and DUXFOR. Lastly, we asked private contacts, matching our target group, to answer the questionnaire. The software companies were contacted multiple times by mail reminding and encouraging them to complete the questionnaire.

Participants: A total of 5 participants from 3 difference software companies were interviewed and acted as pilot testers for our survey. These companies ranged from having 12 to 4.100 employees. A total of 102 companies were contacted regarding participating in this survey, this included private contacts. 67 companies received the questionnaire, 16 companies promised to call back or forward our request for a contact person never did. 19 companies did not want to participant, either because of time or lack of interest in the research.

Through private contacts and posts via social media, Facebook and LinkedIn, the questionnaire reached upwards of 1017 people. Furthermore, the closed Facebook groups reached upwards of 67.394 people with the ability to forward to interested parties.

The respondents were promised anonymity, in order to prevent disclosing information that may be perceived as negative for the organization. The participants were not asked to identify their organization, but rather size and the type of software/service they provide.

A total number of 53 responses were collected. These responses were received from among the 67 companies which received a link and from the sharing on the various social media pages. The forums on which the survey was posted did not contribute with any answers.

Setting: This research was conducted as an online questionnaire, which was finalized and available for the participants online, spanning a period of 12 weeks. The questionnaire was targeted at companies with Danish development branches. The reason for only targeting software development in Denmark was, that it would be possible to describe the current state of customer and end-user inclusion in this country. By limiting it to only one country, it is possible for further research in other countries to be compared to the findings of this study, to gain an idea of the differences between countries.

Data Collection: Initially data were collected through interviews, where the participants were asked about their organization's work practices with regards to customers and end-users. In the pilot questionnaire participants were asked to comment on the questions and describes if there were any discrepancies or ambiguities. The final questionnaire was created in Google Forms and written in Danish. The data collection is the online questionnaire used to describe the current state of customer and end-user participation in companies in the Danish software industry.

Data analysis: The data collected from the online questionnaire were analyzed. This was done by creating tables and graphs of the different data. Furthermore, this includes performing Chi-squared tests in order to identify whether there is a dependency between different answers.

Findings

Respondents demographic

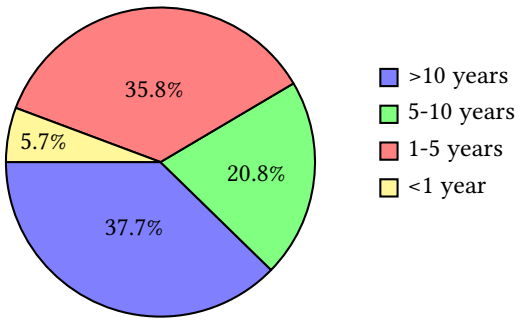

Figure 2. Respondents years of employment at current company

The respondents' were asked about years of employment at their current company and to assess their own expertise. The respondents had 5 different proficiency scales describing expertise to choose between: Fundamental awareness, meaning they have common knowledge or an understanding of basic techniques and concepts [35]. Novice, meaning they have the level of experience gained in a classroom and/or experimental scenarios or as a trainee on-the-job. They are expected to need help when performing this skill [35]. Intermediate, meaning they are able to successfully complete tasks in this competency as requested. Help from an expert may be required from time to time, but they can usually perform the skill independently [35]. Advanced, meaning that they can perform the actions associated with this skill without assistance. They are certainly recognized within their immediate organization as "a person to ask" when difficult questions arise regarding this skill [35]. Or expert, they are known as an expert in this area. They can provide guidance, troubleshoot and answer questions related to this area of expertise and the field where the skill is used [35].

From Figure 2 it can be see that 58.5% of the respondents have worked more than 5 years at the same workplace. From Table 1 we see that 79.2% of the respondents either see themselves as having advanced knowledge or being an expert in their profession.

Expertise	**No (% of total)**
Expert	27 (50.9%)
Advanced	15 (28.3%)
Intermediate	10 (18.9%)
Novice	0 (0.0%)
Fundamental awareness	1 (1.9%)

Table 1. Respondents expertise

Job title	**No (% of total)**
Software/system designer/developer	22 (41.5%)
Owner, manager, supervisory position	16 (30.2%)
IT consultant, consultant	11 (20.8%)
Project managers	12 (22.6%)
Usability expert/architect/designer/engineer	9 (17.0%)
System administrator/engineer/architect/manager	8 (15.1%)
User experience analyst/designer/manager	6 (11.3%)
Designer/UI designer	3 (5.7%)
IT strategist/employee/engineer/investigator	3 (5.7%)
Business analyst/developer	3 (5.7%)
Web designer/master/editor/etc, info master	3 (5.7%)
Interaction designer/architect	1 (1.9%)
PhD students, university teachers, professors	0 (0.0%)
Administrator, investigator	0 (0.0%)
Other	8 (15.1%)

Table 2. Job distribution of respondent (multiple answers possible)

The respondents were asked regarding their current job position and work responsibility. Table 2 shows the job titles the respondents chose. It was possible for the respondents to choose multiple answers, as it is possible for them to

have multiple roles. The Other option includes answers like student programmer, chief of deployment, test automatization, director of engineering, Scrum master and quality/test manager.

Of the 22 Software/system designer/developers 10 of these chose only this as their role, with the remaining have other roles like project manager. Regarding the role of Owner, manager, supervisory position 7 respondents chose this as their only role, where the remaining chose a mix between Software/system designer/developer, IT consultant, System administrator/engineer/architect/manager, Business analyst/developer and Project manager.

Company demographic

The respondents were asked about the company they worked for. Here they were asked regarding the size of the company. The reason for asking this was to get an indication of the companies participating in this survey, and to be able to investigate if there were any correlation between company sizes and the answers the respondents give. The sizes of the companies can be seen in Figure 3.

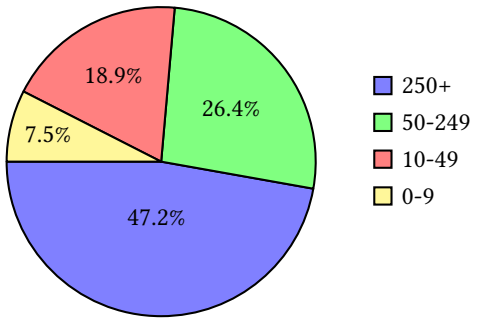

Figure 3. Number of employees in the respondents companies

In Figure 3 it can be seen that 73.6% of the respondents work at companies of 50 or more employees. This was an interesting observation since Danmarks Statistik in 2017 outlines the size of the average Danish software company at around 5 employees (11.214 software related companies with a total of 56.241 employees) [24]. It was possible for the respondents to specify multiple products that their company offers. The reason for asking this question was to get an idea of the frequency of different services and products that are offered. Table 3 shows the different types of product, in order to get a understanding the distribution of business types. It can be seen that more than half of the respondents produce software for either, internal use or custom software or provides Services, API's ect.

Type of business	No (% of total)
Development of internal systems	30 (56.6%)
Development of custom software	28 (52.8%)
Services, API's etc.	27 (50.9%)
Off-the-shelf software	21 (39.6%)
Consultancy	20 (37.7%)

Table 3. Type of business (multiple answers possible)

The respondents were asked to specify the development methods used by their company. The reason for this is to gain an insight into what methods are used in the industry and especially relating to if the methods facilitate customer and end-user participation. Table 4 shows the different methods used by 3 or more of the companies.

It can be seen that primarily Scrum and Kanban are used by the companies. Furthermore, it can be seen that the agile development methods are the most used. The Other option includes methods and techniques like Rapid Application Development, Agile Unified process, Big Design Up Front and Large-Scale Scrum. Out of the 42 respondents that uses Scrum, 29 also uses Kanban as a method they work with. 16 of the Scrum respondents also uses Feature-driven development. 16 uses DevOps and 14 uses SAFe in addition to Scrum.

Development methods	No (% of total)
Scrum	42 (79.2%)
Kanban	29 (54.7%)
Feature-driven development	16 (30.2%)
DevOps	16 (30.2%)
Scaled Agile Framework (SAFe)	14 (26.4%)
Test-driven development	11 (20.8%)
Object-oriented analysis and design (OOAD)	11 (20.8%)
Scrumban	7 (13.2%)
Extreme programming (XP)	3 (5.7%)
Other	15 (28.3%)

Table 4. Development method used by the companies (multiple answers possible)

Methods and techniques used to facilitate customer participation

The respondents were asked to specify the techniques and methods used to facilitate customer participation. A customer is defined as the company / person receiving the developed service / software. Furthermore, the respondents

were asked to specify in which phase of development these techniques and methods were used. Table 5 shows what techniques and methods were used to facilitate customer participation and when they were utilized. It can be seen that the most prominent techniques used during Concept, Analysis and Design phases were Interview, User Stories and Storyboarding. For the Implementation phase Demonstration and Code reviews with the customer was the most used. For the Test phase Usability testing and Observations of real usage were the most prevalent. For Maintenance Observations of real usage, Demonstration and Interview were the most used. It can be seen that in terms of the number of companies that use the techniques and methods Interview and Demonstration were the most used.

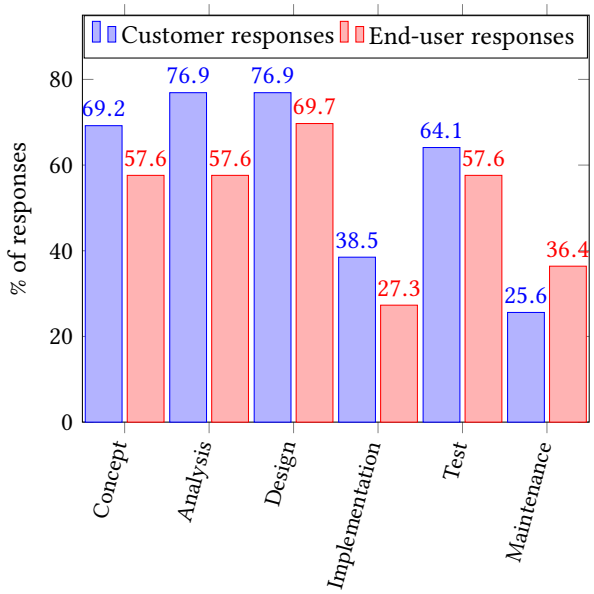

Figure 4. When customer/end-user participation is deem useful by respondents (multiple answers possible)

Methods and techniques used to facilitate end-user participation

To investigate end-user participation, the respondents were asked the same questions regarding end-user participation, as they were asked about customer participation. An end-user is defined as the user, who ultimately uses the product. Table 5 shows the responses regarding the methods and techniques used during development to facilitate end-user participation. It can be seen that for the Concept-, Analysis- and Design phase Interview was the most used technique. Following Interview was User stories, Observations of real usage and Use case analysis. For the Implementation phase Demonstration was the most used technique. For the Testing phase, Usability testing was the most used technique. For the Maintenance phase, Interview and Observations of real usage were the most used techniques to facilitate end-user participation. Similar to techniques used to facilitate customer participation, the most used techniques to facilitate end-user participation were Interview and Demonstration.

An interesting observation was that Demonstration were used by 30 (76.9%) respondents to include customers but only 16 (48.5%) used Demonstration with end-users. Similar to this was the usage of User stories where 24 (61.5%) used User stories for customers and 10 (30%) used it for end-users. Another interesting observation was the usage of prototypes where Lo-fi prototypes were a more popular approach than Hi-fi prototypes, despite many tools exist to help create Hi-fi prototypes. Another observation was that the method Storyboarding was used by 19 (48.7%) respondents when working with customers but only by 6 (18.2%) when working with end-users.

When is customer and end-user participation deemed most useful?

The respondents were asked regrading when they found customer and end-user participation to be useful during development. Figure 4 shows that customer and end-user participation in the Design phase was the highest voted useful time for participation, closely followed by the Analysis- and Concept phases. The Implementation phase was the least useful phase to have the end-user participate and the maintenance phase was the least useful time for customer participation. When looking at the phases with least useful customer and end-user inclusion the used techniques here are primarily Interview and Demonstration. A Chi-squared test was conducted to investigate if the development method used by the company had a correlation as to when customer and end-user participation were deemed useful. The test revealed that there were no significant difference (Customer $p=1.000$, End-user $p=0.997$). An interesting observation was that the difference in responses for the four highest voted phases were at most 12.8% for customers and 12.1 % for end-users.

The respondents were then asked if they thought that customer and end-user participation was sufficient in their company. 11 (28.2%) of the respondents answered that they thought that there was insufficient customer participation in their company. Regarding end-user participation, 17 (51.5%) respondents answered that there was insufficient end-user participation currently in their company. 14 (82.3%) of the

Methods and techniques	**Used in at least one phase**	Number of responds in each phase (% of total)					
		Concept	**Analysis**	**Design**	**Implementation**	**Test**	**Maintenance**
Interview	26 / 23 (66.6%) / (69.7%)	20 / 16 (51.3%)/(48.5%)	18 / 18 (46.2%)/(54.5%)	14 / 13 (35.9%)/(39.4%)	6 / 7 (15.4%)/(21.2%)	9 / 7 (23.1%)/(21.2%)	6 / 8 (15.4%)/(24.2%)
Demonstration	30 / 16 (76.9%) / (48.5%)	12 / 5 (30.7%)/(15.2%)	13 / 5 (33.3%)/(15.2%)	17 / 10 (43.6%)/(30.3%)	23 / 10 (59.0%)/(30.3%)	16 / 8 (41.0%)/(24.2%)	6 / 6 (15.4%)/(18.2%)
User stories	24 / 10 (61.5%) / (30.3%)	15 / 9 (38.5%)/(27.3%)	19 / 8 (48.7%)/(24.2%)	18 / 4 (46.2%)/(12.1%)	14 / 3 (35.9%)/(9.1%)	10 / 3 (25.6%)/(9.1%)	3 / 3 (7.7%)/(9.1%)
Usability testing (Qualitative)	20 / 13 (51.3%) / (39.4%)	6 / 5 (15.4%)/(15.2%)	6 / 7 (15.4%)/(21.2%)	14 / 8 (35.9%)/(24.2%)	12 / 7 (30.8%)/(21.2%)	14 / 10 (35.9%)/(30.3%)	4 / 3 (10.3%)/(9.1%)
Use case analysis	22 / 11 (56.4%) / (33.3%)	14 / 6 (35.9%)/(18.2%)	16 / 9 (41.0%)/(27.3%)	14 / 7 (35.9%)/(21.2%)	7 / 3 (17.9%)/(9.1%)	5 / 1 (12.8%)/(3.0%)	3 / 2 (7.7%)/(6.1%)
Observation of real usages	17 / 14 (43.3%) / (42.4%)	9 / 7 (23.1%)/(21.2%)	10 / 10 (25.6%)/(30.3%)	8 / 8 (20.5%)/(24.2%)	9 / 7 (23.1%)/(21.2%)	10 / 7 (25.6%)/(21.2%)	8 / 8 (20.5%)/(24.2%)
Lo-fi prototyping	14 / 11 (35.9%) / (33.3%)	12 / 9 (30.8%)/(27.3%)	11 / 6 (28.2%)/(18.2%)	9 / 6 (23.1%)/(18.2%)	1 / 1 (2.6%)/(3.0%)	1 / 1 (2.6%)/(3.0%)	1 / 1 (2.6%)/(3.0%)
Storyboarding	19 / 6 (48.7%) / (18.2%)	11 / 3 (28.2%)/(9.1%)	17 / 6 (43.6%)/(18.2%)	11 / 3 (26.2%)/(9.1%)	2 / 3 (5.1%)/(9.1%)	0 / 1 (0.0%)/(3.0%)	0 / 1 (0.0%)/(3.0%)
Code review	21 / 4 (53.8%) / (12.1%)	0 / 1 (0.0%)/(3.0%)	1 / 1 (2.6%)/(3.0%)	3 / 2 (7.7%)/(6.1%)	20 / 3 (51.3%)/(9.1%)	11 / 2 (28.2%)/(6.1%)	6 / 2 (15.4%)/(6.1%)
Questionnaire	13 / 10 (33.3%) / (30.3%)	9 / 5 (23.1%)/(15.2%)	9 / 6 (23.1%)/(18.2%)	2 / 4 (5.1%)/(12.1%)	1 / 1 (2.6%)/(3.0%)	3 / 2 (7.7%)/(6.1%)	2 / 3 (5.1%)/(9.1%)
Style guides	18 / 5 (46.2%) / (15.2%)	8 / 2 (20.5%)/(6.1%)	8 / 3 (20.5%)/(9.1%)	15 / 4 (38.5%)/(12.1%)	11 / 4 (28.2%)/(12.1%)	6 / 2 (15.4%)/(6.1%)	4 / 2 (10.3%)/(6.1%)
Walkthrough	14 / 8 (35.9%) / (24.2%)	7 / 3 (17.9%)/(9.1%)	8 / 2 (20.5%)/(6.1%)	5 / 5 (12.8%)/(15.2%)	6 / 4 (15.4%)/(12.1%)	6 / 5 (15.4%)/(15.2%)	2 / 3 (5.1%)/(9.1%)
Scenarios	15 / 7 (38.5%) / (21.2%)	11 / 6 (28.2%)/(18.2%)	13 / 5 (33.3%)/(15.2%)	9 / 5 (23.1%)/(15.2%)	5 / 3 (12.8%)/(9.1%)	3 / 2 (7.7%)/(6.1%)	1 / 1 (2.6%)/(3.0%)
Focus groups	16 / 6 (41.0%) / (18.2%)	12 / 4 (30.8%)/(12.1%)	8 / 4 (20.5%)/(12.1%)	5 / 2 (12.8%)/(6.1%)	3 / 1 (7.7%)/(3.0%)	2 / 2 (5.1%)/(6.1%)	2 / 1 (5.1%)/(3.0%)
Usage centered design	13 / 8 (33.3%) / (24.2%)	9 / 6 (23.1%)/(18.2%)	10 / 7 (25.6%)/(21.2%)	13 / 6 (33.3%)/(18.2%)	6 / 4 (15.4%)/(12.1%)	6 / 4 (15.4%)/(12.1%)	4 / 4 (10.3%)/(12.1%)
Hi-fi prototyping	12 / 8 (30.8%) / (24.2%)	4 / 3 (10.3%)/(9.1%)	7 / 2 (17.9%)/(6.1%)	11 / 7 (28.2%)/(21.2%)	4 / 3 (10.3%)/(9.1%)	1 / 4 (2.6%)/(12.1%)	1 / 1 (2.6%)/(3.0%)
UI stories	15 / 5 (38.5%) / (15.2%)	10 / 3 (25.6%)/(9.1%)	10 / 4 (25.6%)/(12.1%)	9 / 3 (23.1%)/(9.1%)	4 / 2 (10.3%)/(6.1%)	2 / 1 (5.1%)/(3.0%)	1 / 1 (2.6%)/(3.0%)
Quantitative usability testing	13 / 5 (33.3%) / (15.1%)	3 / 4 (7.7%)/(12.1%)	2 / 3 (5.1%)/(9.1%)	5 / 4 (12.8%)/(12.1%)	10 / 1 (25.6%)/(3.0%)	11 / 3 (28.2%)/(9.1%)	7 / 2 (17.9%)/(6.1%)
Persona	12 / 5 (30.8%) / (15.1%)	9 / 4 (23.1%)/(12.1%)	10 / 4 (25.6%)/(12.1%)	9 / 2 (23.1%)/(6.1%)	6 / 1 (15.4%)/(3.0%)	4 / 1 (10.3%)/(3.0%)	1 / 1 (2.6%)/(3.0%)
Contextual analysis	9 / 4 (23.1%) / (12.1%)	9 / 2 (23.1%)/(6.1%)	9 / 4 (23.1%)/(12.1%)	5 / 1 (12.8%)/(3.0%)	2 / 0 (5.1%)/(0.0%)	1 / 0 (2.6%)/(0.0%)	1 / 0 (2.6%)/(0.0%)
Competitive analysis	9 / 4 (23.1%) / (12.1%)	7 / 2 (17.9%)/(6.1%)	8 / 4 (20.5%)/(12.1%)	4 / 4 (10.3%)/(12.1%)	2 / 1 (5.1%)/(3.0%)	1 / 1 (2.6%)/(3.0%)	2 / 2 (5.1%)/(6.1%)
Expert or heuristic evaluation	7 / 5 (17.9%) / (15.2%)	5 / 4 (12.8%)/(12.1%)	4 / 4 (10.3%)/(12.1%)	4 / 2 (10.3%)/(6.1%)	4 / 2 (10.3%)/(6.1%)	4 / 2 (10.3%)/(6.1%)	2 / 2 (5.1%)/(6.1%)
Setting quantitative usability goals	7 / 3 (17.9%) / (9.1%)	6 / 1 (15.4%)/(3.0%)	5 / 2 (12.8%)/(6.1%)	3 / 2 (7.7%)/(6.1%)	1 / 0 (2.6%)/(0.0%)	1 / 1 (2.6%)/(3.0%)	1 / 0 (2.6%)/(0.0%)
Cognitive walkthrough	5 / 2 (12.8%) / (6.1%)	4 / 2 (10.3%)/(6.1%)	4 / 1 (10.3%)/(3.0%)	2 / 1 (5.1%)/(3.0%)	2 / 1 (5.1%)/(3.0%)	1 / 1 (2.6%)/(3.0%)	2 / 1 (5.1%)/(3.0%)
Early human factors analysis	4 / 2 (10.3%) / (6.1%)	3 / 2 (7.7%)/(6.1%)	4 / 1 (10.3%)/(3.0%)	2 / 1 (5.1%)/(3.0%)	1 / 1 (2.6%)/(3.0%)	1 / 1 (2.6%)/(3.0%)	1 / 1 (2.6%)/(3.0%)

Table 5. Methods and techniques used by companies to facilitate customer / end-user participation. The total number of answers of customer participation were 39 and end-user participation were 33

respondents, that thought that there was insufficient end-user participation, used Scrum as a development method. Another observation was that end-users were deemed more useful in the maintenance phase than customers. For this phase end-user participation was ensured through Interview and Observations of real usage.

Chi-squared tests were conducted to investigate if the type of software a company produced, had a correlation to the

amount of customer and end-user participation. The test results showed no significant difference between type of product and participation(Customer p=0.745, End-user p=0.956). Furthermore, a Chi-squared test was conducted to investigate a possible correlation between a respondents job title and if the amount of customer and end-user participation were sufficient. This test revealed that there were no significant difference (Customer p=0.068, End-user p=0.226). In addition to this another Chi-squared test revealed that there was no significant difference between the development methods used by a company and if the amount of customer and end-user participation were sufficient (Customer p=0.384, End-user p=0.804). The respondents were asked to evaluate the current level of customer and end-user participation on a scale from 1-7 where 1 was no contact with customer/end-user and 7 was that the customer/end-user was deeply participating in the appropriate phases of development. Regarding evaluating customer participation 27 (69%) of the respondents evaluated their current customer participation at 5 or above. In relation to the remaining respondents and the size of the company they work for, 3 (25%) of those respondents worked for companies with 50-249 employees and 7 (58.3%) of them worked for companies with 250+ employees. This meant that 83.3% of responses rating 4 or below were from companies with 50 or more employees. When looking at current end-user participation 15 (46%) of the respondents evaluated their participation at 5 or above. In relation to those that answered 4 or below regarding end-user participation, 12 (66.6%) of the respondents work at a company with 50 or more employees.

A Chi-squared test revealed that there was no significant difference between the type of software the company creates and the evaluated level of customer and end-user participation in the company (Customer p=0.584, End-user p=0.990). Furthermore, when investigating if the size of the company had a correlation with the evaluated level of customer and end-user participation, the Chi-squared test showed that there was no significant difference (Customer p=0.763, End-user p=0.745). Another Chi-squared test was conducted to investigate if the techniques, shown in Table 5, used to include the customer and end-user had an effect on the estimated level of customer and end-user participation. This revealed no significant difference (Customer p=0.993, End-user p=0.997), meaning that the techniques used to include the customer and end-user had no significant correlation to the estimated levels.

What obstacles are there for more customer and end-user participation?

We asked the respondents what obstacles that prevented further customer and end-user participation during the development they were involved in. Table 6 shows that the geographic location of both the customer and end-users was one of the most prominent obstacles. Furthermore, additional customer participation was also deemed too time consuming, where the obstacles for additional end-user participation were more varied. The Other option includes answers with >3 votes, some of these were; It was too expensive, that the customer was only interested in receiving a working product not how it got there, that it was hard to find a representative group of end-users and that the respondents were afraid that by utilizing the end-users they would give the end-users the wrong expectations of the system.

Obstacles	**Customer No (% of total)**	**End-user No (% of total)**
To time consuming	13 (33.3%)	8 (24.2%)
Geographic challenges	12 (20.8%)	9 (27.3%)
Lack of staff	10 (25.6%)	5 (15.2%)
Not a company priority	5 (12.8%)	7 (21.2%)
Does not fit development process	4 (10.3%)	7 (21.2%)
Lack of expertise	6 (15.4%)	5 (15.2%)
To expensive	5 (12.8%)	5 (15.6%)
Customer's lack of expertise or resources	4 (10.3%)	0 (0.0%)
Other	8 (20.5%)	12 (36.4%)

Table 6. Obstacles regarding further customer/end-user participation (multiple answers possible)

Looking at the obstacles in relation to the development methods used by the companies. It can be seen that all except one of the respondents that have specified that, participation of customers/end-users does not fit their development process, have answered Scrum as one of their development processes. The respondents that chose Time as an obstacle had chosen Scrum and Kanban as their development methods, with a few also choosing Feature-driven development. The Geographical challenges, accounts for 20.8% and 27.3%, of customer and end-users respectively, out of the total responses. Geographical challenges include multiple things, since the respondents were not able to choose between distance, time zone nor social cultural differences, therefor all these are part of Geographical challenges. Despite this, a Chi-squared test revealed that there were no significant difference between the development methods used by a company and the obstacles they faced (Customer p=0.935, End-user p=0.923). Furthermore, when looking at the respondents that answered that

Lack of staff were a challenge, it showed that 8 of the 10 that answers this for customer were working for a company with 50 or more employees. In addition to this all that answered Lack of staff for end-user participation were working for a company with 50 or more employees.

To further investigate if obstacles had a correlation with the type of software or size of a company Chi-squared tests were conducted. In relation to obstacles they faced and type of software they produced the Chi-squared test showed no significant difference (Customer p=1.000, End-user p=0.964) and no significant difference between obstacles they faced and the size of the company (Customer p=0.169, End-user p=0.354).

Discussion

This section discusses interesting observations made during the research. Followed by the results discussed in relation to related work. Lastly the practical reality of the results is discussed.

An interesting observation made during this research was the low number of responses received. Some of the challenges, that is known to have an impact on the number of responses is relevance for the respondent, length of the questionnaire, how the survey is delivered to the respondent and the incentive for answering. To overcome the challenges of relevance, a large effort was made to contact companies which had a software development branch in Denmark. A total of 102 companies were called directly by the authors and asked if they would be willing to participate in this survey. In addition to this, various social media website and forums were used, to potentially reach upwards of 67.000 people, to gain additional respondents for the survey. It was ensured that the groups which the questionnaire was shared in also consisted of people with software development knowledge. To account for the challenge of questionnaire length, we followed Fan and Yan [28], which explains that the length of the survey should be kept short to around 10 minutes. Additionally, in terms of the questionnaires' delivery, we directly called companies to avoid the challenge of spam filters and by that ensuring that only the companies who agreed to participated received the questionnaire. Furthermore, we added an incentive to answering the survey which was that all collected data would be available for the respondents if they wanted it.

In regards to related work, Bak et al. [2] showed that the most prominent obstacles for further customer and end-user participation were perceived resource demands and the mindset of developers. This research had similar findings, which showed that geographic challenges, time, lack of staff and the qualification of staff were limiting factors. The primary identified obstacle for this research was time, which was similar to the obstacle of perceived resource demands identified by Bak et al. [2].

Gulliksen et al. [31] showed that management support and project management support was essential for enabling the usability worker. Furthermore, Gulliksen et al. [31] showed that at the time of their research, user participation was a low priority by companies. In relation to the findings of this research, it can be seen that customer and end-user participation not being a priority by companies is still an obstacle, with 12.8% and 21.2% respectively mentioning this as an obstacle.

Through 12 reports spanning from 2006-2017 CollabNet VersionOne [19] showed which agile development methods that are being used worldwide. Here the most prominent method was Scrum used by 56% in 2017, increased by 16% from 40% in 2006. Similar in our results, Scrum is the most prominent agile development method with 79.2%. In relation to the report from CollabNet VersionOne, the difference in 23.2% can be explained by the respondents only being able to answer one method in the survey from CollabNet VersionOne, while being able to answer multiple development methods in our survey. Furthermore, it can be seen in the survey by CollabNet, that the popularity of hybrid Scrum methods are increasing, which is also supported by our results since most respondents did not use Scrum excursively. For an example, every respondent that choose Kanban as their development method also choose Scrum.

In relation to practice, this paper presented several results regarding how companies in the Danish software industry facilitates customer and end-user participation in their development process. This research can be used to gain an insight into what techniques are used by companies in the industry and when in the development these techniques are most prevalent. This can be used by other companies to gain knowledge about the most prevalent techniques, that could be adopted by them. Furthermore, this research highlights some of the obstacles, that exist in the industry, hindering further customer and end-user participation. This can contribute to further research in how these obstacles can be minimized or avoided allowing for further customer and end-user participation.

Conclusion

This paper presented findings that showed the current state of customer and end-user participation in companies in the Danish software industry. This paper showed the methods

and techniques used by companies in the industry to facilitate customer and end-user participation, and in what phases they are most popular. It was discovered that Scrum was the dominating development method, since 79.2% of respondents used it. Furthermore, this paper showed that 51.5% of respondents thought that there were insufficient end-user participation in their current development process and 28.2% thought that there was insufficient customer participation. The most used techniques for facilitating participation were Interview, Demonstration, User stories and Qualitative usability testing, although it was discovered that Demonstration were used by 30 (76.9%) respondents to include customers but only 16 (48.5%) used Demonstration with end-users. Similar to this was the usage of User stories where 24 (61.5%) used User stories for customers and 10 (30%) used it for end-users. The primary identified obstacles for additional customer and end-user inclusion were Lack of time, Geographical challenges and Lack of staff.

We acknowledge the limitations of our work regarding the number of responds to the questionnaire. Hence, the limited number have an effect on the data collected and limit the possibility of making generalized conclusions. Furthermore, some questions allowed multiple answers making it difficult to make connections between different answers.

Additional studies should be performed, investigating the reasoning behind the answers given by the respondents. It would be possible to further investigate the obstacles and how to minimize or avoid these. Furthermore, it would be possible to conduct additional studies, to investigate if the techniques used for both customer and end-user participation are used in different ways depending on the type of participation.

Acknowledgments

We are very grateful to the respondents that helped us by answering our survey. We would like to thank the 5 people that help pilot testing our survey. We would also like to thank those who help distribute our survey either through social media or through internal company emails.

References

[1] P Abrahamsson. 2008. Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. *IET Software* 2, 1 (2 2008), 58–64. https://digital-library.theiet.org/content/journals/10.1049/iet-sen_20070038

[2] Jakob Otkjær Bak, Kim Nguyen, Peter Risgaard, and Jan Stage. 2008. Obstacles to Usability Evaluation in Practice: A Survey of Software Development Organizations. In *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges (NordiCHI '08)*. ACM, New York, NY, USA, 23–32. https://doi.org/10.1145/1463160.1463164

[3] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. 2001. *Manifesto for Agile Software Development*. Technical Report. http://agilemanifesto.org/

[4] A Begel and N Nagappan. 2007. Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*. 255–264. https://doi.org/10.1109/ESEM.2007.12

[5] John Callahan and Eytan Lasry. 2004. The importance of customer input in the development of very new products. *R&D Management* 34, 2 (3 2004), 107–120. https://doi.org/10.1111/j.1467-9310.2004.00327.x

[6] Pilar Carbonell, Ana I Rodríguez-Escudero, and Devashish Pujari. 2009. Customer Involvement in New Service Development: An Examination of Antecedents and Outcomes*. *Journal of Product Innovation Management* 26, 5 (9 2009), 536–550. https://doi.org/10.1111/j.1540-5885.2009.00679.x

[7] CollabNet VersionOne. 2006. *1st annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-c-473508-state-of-agile-report

[8] CollabNet VersionOne. 2007. *2rd annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-338592876-2nd-annual-state-of-agile-report/473508

[9] CollabNet VersionOne. 2008. *3th annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-338592873-3rd-annual-state-of-agile-report/473508

[10] CollabNet VersionOne. 2009. *4th annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-338592849-4th-annual-state-of-agile-report/473508

[11] CollabNet VersionOne. 2010. *5th Annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-338592825-5th-annual-state-of-agile-report/473508

[12] CollabNet VersionOne. 2011. *6th Annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-338592807-6th-annual-state-of-agile-report/473508

[13] CollabNet VersionOne. 2012. *7th Annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-338592786-7th-annual-state-of-agile-report/473508

[14] CollabNet VersionOne. 2013. *8th Annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-338592780-8th-annual-state-of-agile-report/473508

[15] CollabNet VersionOne. 2014. *9th Annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-338592759-9th-annual-state-of-agile-report/473508

[16] CollabNet VersionOne. 2015. *10th Annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-338498988-10th-annual-state-of-agile-report/473508

[17] CollabNet VersionOne. 2016. *11th Annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-338501295-11th-annual-state-of-agile-report/473508

[18] CollabNet VersionOne. 2017. *12th Annual State of Agile Report.* Technical Report. https://www.stateofagile.com/#ufh-i-427987332-12th-annual-state-of-agile-report-overview/473508

[19] CollabNet VersionOne. 2019. DevOps and Agile Solutions for the Enterprise | CollabNet VersionOne. https://www.collab.net/

[20] Colleen Cook, Fred Heath, and Russel L Thompson. 2000. A Meta-Analysis of Response Rates in Web- or Internet-Based Surveys. *Educational and Psychological Measurement* 60, 6 (2000), 821–836. https://doi.org/10.1177/00131640021970934

[21] MICK P COUPER. 2000. Review: Web Surveys: A Review of Issues and Approaches. *Public Opinion Quarterly* 64, 4 (2000), 464–494. https://doi.org/10.1086/318641

[22] Mick P Couper, Arie Kapteyn, Matthias Schonlau, and Joachim Winter. 2007. Noncoverage and nonresponse in an Internet survey. *Social Science Research* 36, 1 (2007), 131–148. https://doi.org/10.1016/j.ssresearch.2005.10.002

[23] Steve D. Fraze, Kelly K. Hardin, M Todd Brashears, Jacqui L. Haygood, and James H. Smith. 2003. The Effects Of Delivery Mode Upon Survey Response Rate And Perceived Attitudes Of Texas Agri-Science Teachers. *Journal of Agricultural Education* 44 (2003), 27–37. https://doi.org/10.5032/jae.2003.02027

[24] Danmarks Statistik. [n. d.]. It-servicevirksomhed - Danmarks Statistik. https://www.dst.dk/da/Statistik/emner/erhvervslivets-sektorer/serviceerhverv/it-servicevirksomhed-serviceydelser

[25] D A Dillman. 2007. *Mail and internet surveys: the tailored design method.* Wiley. https://books.google.dk/books?id=it64AAAAIAAJ

[26] Don A Dillman and Jolene D Smyth. 2007. Design Effects in the Transition to Web-Based Surveys. *American Journal of Preventive Medicine* 32, 5, Supplement (2007), S90 – S96. https://doi.org/10.1016/j.amepre.2007.03.008

[27] Kieren Diment and Sam Garrett-Jones. 2007. How Demographic Characteristics Affect Mode Preference in a Postal/Web Mixed-Mode Survey of Australian Researchers. *Social Science Computer Review* 25, 3 (2007), 410–417. https://doi.org/10.1177/0894439306295393

[28] Weimiao Fan and Zheng Yan. 2010. Factors affecting response rates of the web survey: A systematic review. *Computers in Human Behavior* 26, 2 (2010), 132–139. https://doi.org/10.1016/j.chb.2009.10.015

[29] Kjell Grønhaug, Trond Hammervoll, Sven A Haugland, and Mons Freng Svendsen. 2011. Marketing strategy and customer involvement in product development. *European Journal of Marketing* 45, 4 (4 2011), 513–530. https://doi.org/10.1108/03090561111111316

[30] Jan Gulliksen, Bengt Göransson, Inger Boivie, Stefan Blomkvist, Jenny Persson, and Åsa Cajander. 2003. Key principles for user-centred systems design. *Behaviour & Information Technology* 22, 6 (2003), 397–409. https://doi.org/10.1080/01449290310001624329

[31] Jan Gulliksen, Jan, Boivie, Inger, Persson, Jenny, Hektor, Anders, Herulf, and Lena. 2004. Making a difference: a survey of the usability profession in Sweden. https://doi.org/10.1145/1028014.1028046

[32] Richard Gunther, Jeff Janis, and Scott Butler. [n. d.]. The UCD Decision Matrix: Ovo Studios. http://www.ovostudios.com/upa2001/surf.htm

[33] Maarit Laanti, Outi Salo, and Pekka Abrahamsson. 2011. Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology* 53, 3 (3 2011), 276–290. https://doi.org/10.1016/J.INFSOF.2010.11.010

[34] Xiaohua Lin and Richard Germain. 2004. Antecedents to Customer Involvement in Product Development:: Comparing US and Chinese Firms. *European Management Journal* 22, 2 (2004), 244–255. https://doi.org/10.1016/j.emj.2004.01.009

[35] National Institutes of Health. [n. d.]. Competencies Proficiency Scale | Office of Human Resources. https://hr.nih.gov/working-nih/competencies/competencies-proficiency-scale

[36] Dan R. Vijayasarathy, Leo Turk. 2008. Agile software development: A survey of early adopters. *Journal of Information Technology Management* 19 (5 2008).

[37] O Sohaib and K Khan. 2010. Integrating usability engineering and agile software development: A literature review. In *2010 International Conference On Computer Design and Applications*, Vol. 2. 2–32. https://doi.org/10.1109/ICCDA.2010.5540916

B | DIAbetes application

B.1 Purpose

An application was created as part of the case study of contribution 1. The goal of the application is to providing information and management tools and by that empower the diabetic into having a better understanding of their diabetes. The idea is that the diabetic will be able to take better actions that can positively affect them and their treatment. The application was built around the concepts of patient empowerment, meaning that all implemented functionally were made to provide a way to empower the users of the application.

B.2 Development process

The application was developed following the adapted development method described in contribution 1 in section 2.1. The approach to developing the application was to, initially through the information gathering of the pre-sprint, gain a knowledge base of basic functionality for the application. Then, through the first sprint, refine the identified functionality and have the Product Owner prioritize these. Then through each following sprint, the application was presented to the end-users, which had the opportunity of providing feedback on the application and suggest possible new functionality. The feedback and suggested functionality would be converted into possible item in the Product Backlog that would be prioritized by the Product Owner and implemented by the developers.

The development of the application spanned a period of 8 weeks. Each week had around 30 hours per developer dedicated to working on the application. This meant that the total number of man hours dedicated to the activities required to develop the application was roughly 240 hours per person spanning the entire development. This also included the activities performed with the end-users such as Interviews and Walkthroughs. When performing the remote asynchronous usability tests, the application was continuously updated over a period of 14 days. For this period approximately 4 hours was spent developing the updates for the application each day. The Github project can be found at the following link:

https://github.com/Jakobx99/P9Project

The application is an android application developed for Android SDK 23 (6.0 Marshmallow). The application itself contains a total of 11.634 lines of code written in Java.

B.3 Functionality of the application

In the following section the different pages and functionality of the application will be explained.

B.3.1 Insulin calculator page

To support an activity performed daily by the diabetics, a page was created to help calculate the amount of insulin they need to take to account for their carbohydrate intake. This page can be seen in Figure B.1. This page have a functionality enabling the user to enter the amount of fiber in their diet. If the amount of fiber exceeds 30% of the carbohydrate intake, a message will be displayed to the user with additional information recommending when to take their insulin. This page will automatically fetch the last entered blood glucose measurement entered by the user. Furthermore, all calculations are based on personal values, such as insulin sensitivity, entered on the Profile page.

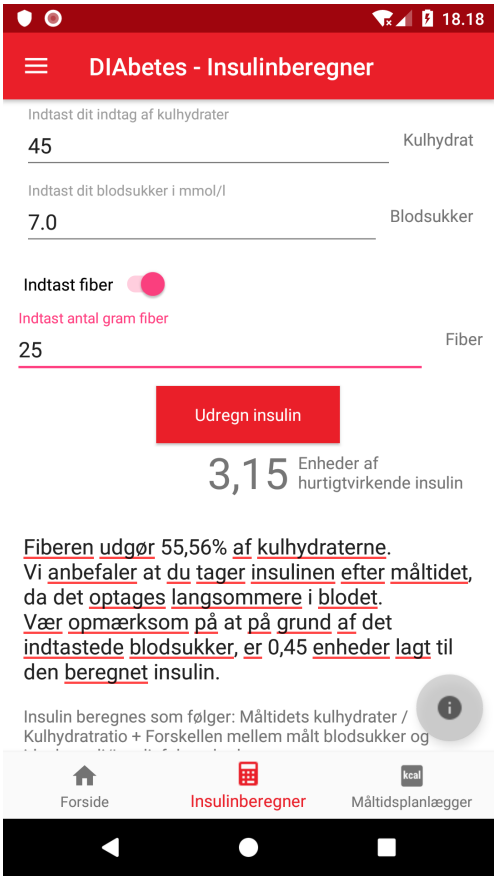

Figure B.1: The Insulin Calculator page

B.3.2 Meal Planner page

The Meal Planner page was created to assist the diabetics with their diet. The purpose of the meal planner page was to provide information regarding carbohydrates and fiber of the food the diabetics eat. Furthermore, by entering a meal in the Meal Planner it is able to calculate the amount of insulin needed for the meal. The Meal Planner page can be seen in Figure B.2. Similar to the Insulin Calculator the Meal Planner will also automatically fetch the latest blood glucose measurement and the personal information from the Profile page. All meals calculated with the Meal Planner is saved and can be found in a list that contains all previous meals. The list can be seen in Figure B.3. By selecting a previous meal the users will be presented with a page similar to the Meal Planner page showing the previous meal.

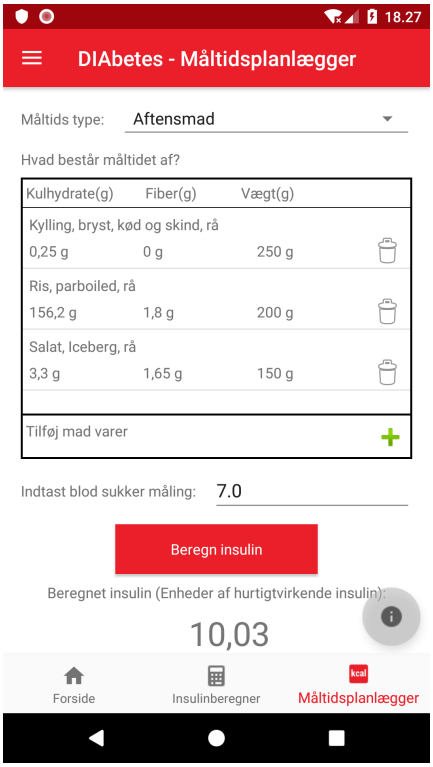

Figure B.2: The Meal Planner page

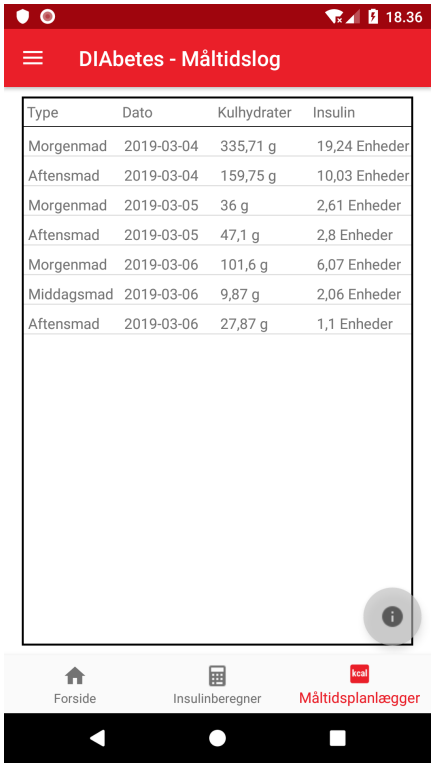

Figure B.3: The List of previous meals page

B.3.3 Blood glucose measurement pages

A core functionality of the application is to enable the users to track their blood glucose measurements. To facilitate this, multiple pages have been created to ensure that the users have as much information available as possible.

New Measurement page: A page has been create enabling the users to enter a blood glucose measurement. This page requires the user to specify a date and time for the measurement, the measurement itself and if it was related to a meal. The application will automatically fill in the date and time but allows the user to change it if the measurement is from a different time. This page can be seen in Figure B.4.

Overview page: A page has been create to provide an overview of the blood glucose measurements. On the top of the page is a small graph illustrating the various measurements that have been made during the current week. The colors used are based on whether the measurements are below, above or between the allowed lower and upper limits. These limits are entered by the user on the Profile page. In the middle of the page is the value of the latest measurement. The border of the text-box containing the value is colored to help identify if this measurement were too low or high. At the bottom of the page is three colored boxes

displaying the number of low, high or between measurements made in the current week. The sizes of the boxes change depending on the relation between the number of measurements. These boxes were made to help give the user a quick overview to help inform them if they have been able to regulate their diabetes correctly. This page can be seen in Figure B.5.

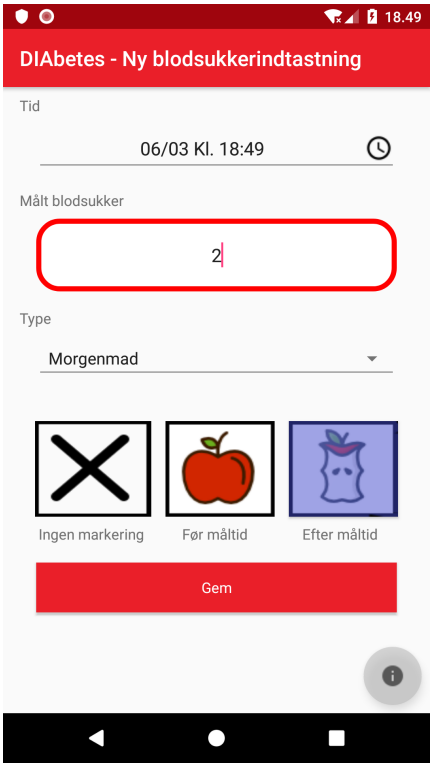

Figure B.4: The New Measurement page

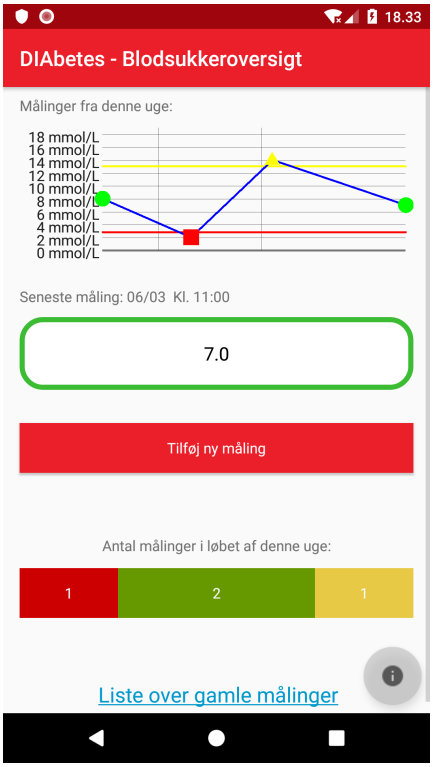

Figure B.5: The Overview page

Graph page: If the graph on the Overview page is clicked, it will redirect to another page where the graph fills the entire page. The purpose of this page is to provide a visual representation of all measurements spanning a specific time-period. The default time-period is the current week, but by clicking the gear icon the user can specify the start and end date displayed on the graph. The graph contains colored lines to mark the separations of low, high and in-between measurements. This page can be seen in Figure B.6. In addition to this if the user has entered a long-term blood glucose measurement, taken by their physician, this measurement is also displayed with the black line on the graph. This helps inform the user if their daily measurements are close to their average which is displayed with the long-term measurement.

List of old blood glucose measurements: The application saves all entered blood glucose measurements and in addition to the Graph page, the application have another page that displayed previous measurements. This page provides a list with all previous measurements

spanning a specified time-period. All entries in the list are color coded to help inform the user about the measurements. This page can be seen in Figure B.7. The user is able to change the time-period of the measurements displayed in the list.

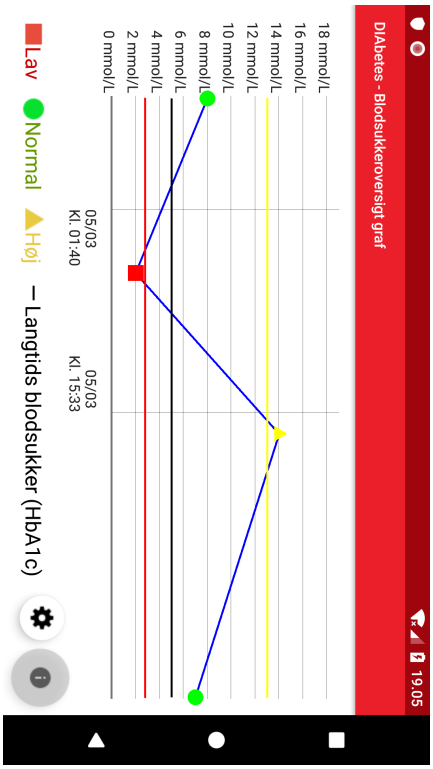

Figure B.6: The Graph page

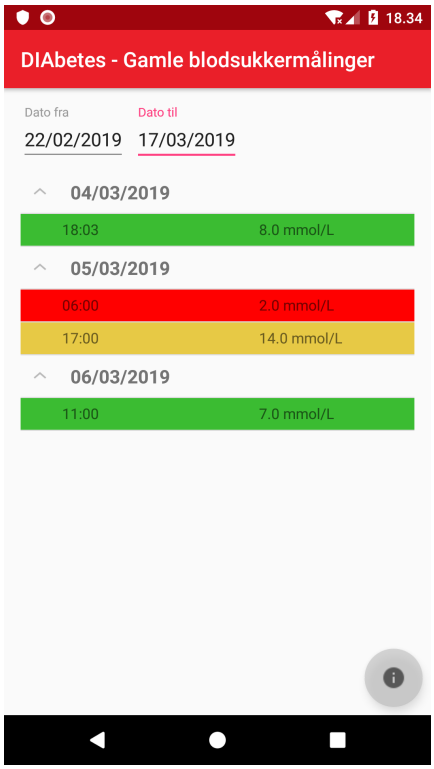

Figure B.7: The old blood glucose measurements page

B.3.4 Profile page

A profile page have been created to ensure that the correct information is used about the user. On the profile page the user enters: their target blood glucose level, daily insulin usage, carbohydrate ratio, insulin sensitivity and the upper and lower bounds for their ideal blood glucose levels. There is a relationship between daily insulin usage, carbohydrate ratio and insulin sensitivity. When the user fills in one of these three, the two others are automatically calculated. The profile page can be seen in Figure B.8.

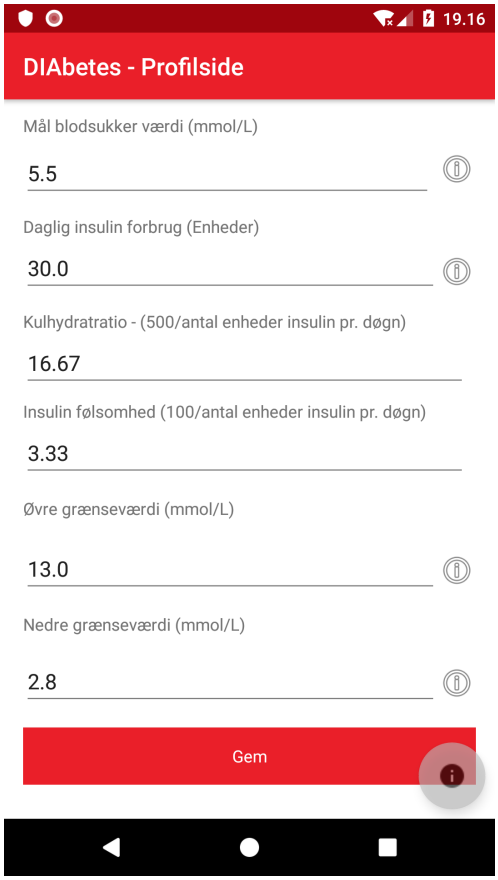

Figure B.8: The Profile page

B.3.5 Notes page

It is possible to create notes in the application. This functionality was create based on diabetics telling us that normally they write down different question for their physician so they can ask these at the next consultation. Having this functionality in the application helps to facilitate this aspect of a diabetics daily life.

B.3.6 FAQ page

A page was create that contains frequently asked questions. This page was create to ensure that relevant and useful information was available in the application. This functionality was create based on users expressing that they often had question regarding how to account for diabetes during many of their daily activities. The FAQ page enables the user to select different categories of questions and to search for specific key words in questions. The FAQ

page can be seen in Figure B.9.

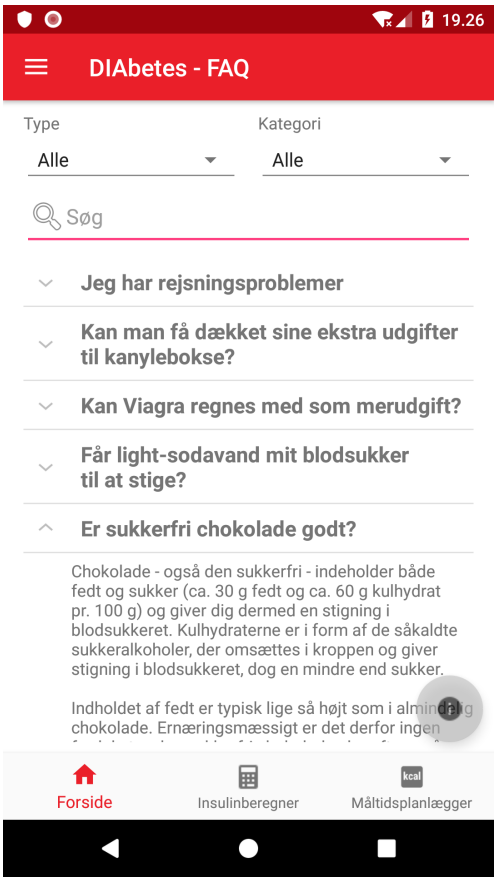

Figure B.9: The FAQ page

B.3.7 Parent supervision functionality

During development several users mentioned that they would like a parental supervision functionality. A page was created that enabled a user to enter a phone number and enable parental supervision. By enabling this, every time an insulin calculation was made the information of this calculation, such as blood glucose level and recommended insulin, was sent via a SMS to the entered phone number. In addition to this every time a new blood glucose measurement was create in the application a SMS with the details for this was also sent to the entered phone number.

B.3.8 Generate PDF page

Several users mentioned that they would like the ability to get a PDF document of their blood glucose measurements so they would be able to show these to their physician. A page were created supporting the functionality of generating a PDF file that contains blood glucose measurements for the last three months together with a graph of these. The user have the ability to write an email address on the Generate PDF page, which will then, in addition to saving the PDF on the phone, also send it to the entered email address. An example of what such a PDF file could contain can be seen in Figure B.10.

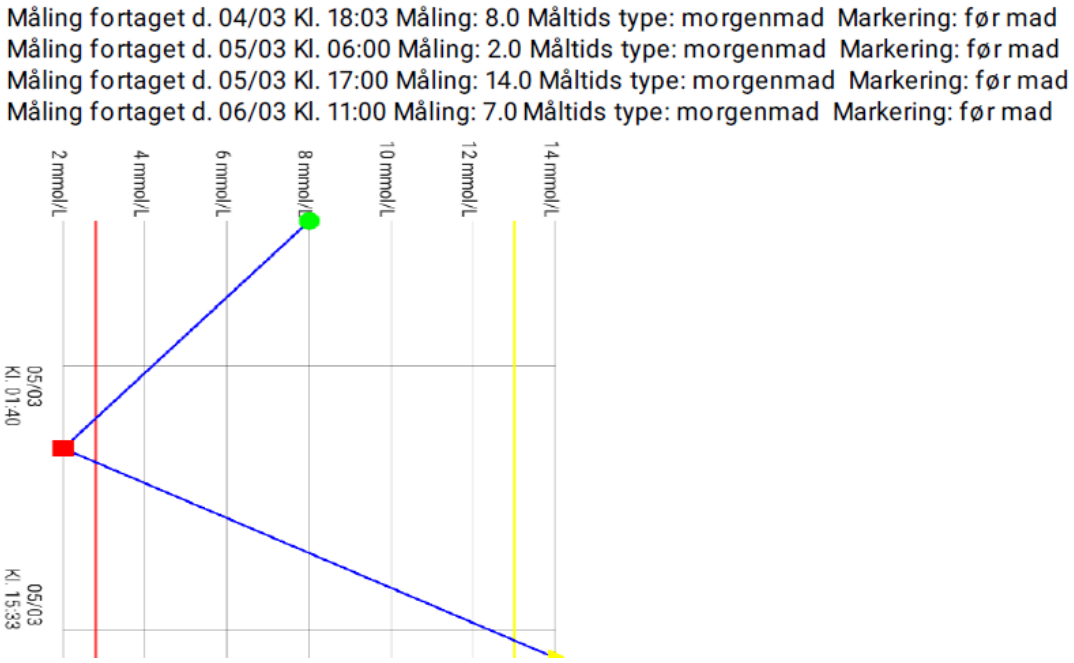

Figure B.10: A generated PDF