The UXObserver

Design of a usability testing tool for Windows tablets

Sofie Eliasson
ic08se8@student.lth.se

Jessika Nilsson
ic07jn0@student.lth.se

January 31, 2014

Abstract

How can usability testing be improved? Is there a way to simplify the gathering of data without taking notes with a pen and paper and can this be done by using a tablet? These are the questions we asked ourselves in the beginning of the master thesis project. This report describes the development process from idea to final prototype.

The project has focused on user centered design which included a close collaboration with employees which regularly perform usability testing. This has been carried out iteratively with recurrent interviews and testing sessions in the different stages of the development.

The final testing confirmed that test participants could use the prototype when testing instead of taking notes. All of the participants expressed positive interests to the product and felt that they would like to use it in the future when it has been finalised and some functionality been added. They also believed it could save them time in the process of planning, performing and analysing usability tests. The conclusion is that a test leader can perform usability testing while using a tablet, but that our product is still not finished. The prototype needs to be adjusted and functions added, as well as become complemented with a computer program for more advanced functions.

Sammanfattning

Hur kan användbarhetstester effektiviseras? Kan insamlingen av data förenklas genom att utesluta traditionella hjälpmedel som penna och papper och istället använda sig av en surfplatta? Dessa frågor ställde vi oss i början av examensarbetet. Rapporten beskriver utvecklingsprocessen från idé till färdig prototyp.

Examensarbetet har fokuserat på användarcentrerad design som inkluderat ett nära samarbete med personer som regelbundet utför användbarhetstester. Detta har skett iterativt med återkommande intervjuer och testsessioner i olika stadier av utvecklingen.

Sluttestet bekräftade att testpersonerna kunde använda prototypen vid test i stället för att ta anteckningar med papper och penna. Samtliga deltagare uttryckte ett positivt intresse för produkten och de skulle vilja använda den i framtiden när den har färdigställts och önskad funktionalitet lagts till. De ansåg även att det skulle kunna spara dem tid i processen för planering, genomförande och analys av användbarhetstester. Slutsatsen är att en testledare kan utföra användartester genom användning av en surfplatta, men att vår produkt fortfarande kräver viss utveckling. Prototypen behöver justeras och funktioner läggas till samt bli kompletterad med ett datorprogram för mer avancerade funktioner för analys av den insamlade datan.

Acknowledgements

We would like to give our gratitude to Cenito Software AB and a special thanks to Olis Olofsson for the guidance and support during the application development.

Furthermore we would like to thank Bengt Larsson and Anita Stenquist at Gambro Lundia AB for the support and participation in the interviews and testing during the project.

Thank you to our families, friends, acquaintances and test participants for the support and interest in our work.

Last but not least, a big thank you to Kirsten Rassmus-Gröhn, our supervisor at the Department of Design Sciences at Lund University, for the guidance and input on our work during this master thesis project.

Contents

1 Introduction **6**
- 1.1 Background . . . 6
- 1.2 Purpose and goal . . . 7
- 1.3 Limitations . . . 7
- 1.4 Target Audience . . . 8
- 1.5 Cenito Software AB . . . 8
- 1.6 Division of labor . . . 8

2 Theory **10**
- 2.1 Usability . . . 10
- 2.2 Usability Testing . . . 11
- 2.3 User Centered Design . . . 12
- 2.4 Donald Norman design principles . . . 12
- 2.5 Interaction patterns and measurements . . . 14

3 Method **18**
- 3.1 Feasibility study . . . 18
  - 3.1.1 Literature . . . 18
  - 3.1.2 Interview . . . 18
  - 3.1.3 Observation . . . 19
  - 3.1.4 Previous work in the field . . . 19
- 3.2 Conceptual model . . . 20
- 3.3 Software development . . . 20
  - 3.3.1 Low fidelity prototype . . . 21
  - 3.3.2 High fidelity prototype . . . 21
  - 3.3.3 Code development . . . 25
  - 3.3.4 Final testing . . . 25
- 3.4 Tools . . . 26

4 Result **28**
4.1 Feasibility study 28
4.1.1 Literature 28
4.1.2 Interviews 29
4.1.3 Observation 31
4.1.4 Previous work in the field 32
4.1.5 List of functions 33
4.2 Conceptual design 34
4.2.1 System 34
4.2.2 Mental models and metaphors 35
4.2.3 Requirement specification /Backlog 35
4.3 Development 36
4.3.1 Low fidelity prototype 37
4.3.2 High fidelity prototype 38
4.3.3 Code development 57

5 Walkthrough of prototype version 1 **62**

6 Walkthrough of prototype version 2 **70**

7 Discussion and Conclusion **76**
7.1 Design decisions 76
7.2 Fulfillment of aims 77
7.3 Further development 78
7.4 Conclusion 80

References **84**

8 Attachments / Bilagor: **88**
8.1 Interview questions 1 88
8.2 Observation questions 89
8.3 Data gathered 90
8.3.1 Performance data 90
8.3.2 Preference data 91
8.4 Test documents 92
8.4.1 Task list, test 1 92
8.4.2 Testplan, final testing 94
8.4.3 Task list, test 2 95
8.5 Source of icons 97

Chapter 1

Introduction

1.1 Background

Usability is a term often discussed when e.g. comparing the latest software in the mobile industry, but usability is often difficult to define and means different things to different people. ISO 9241 defines usability as "The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments."[1]. With larger focus on usability and how products should perform and be used, a more specific need for testing has emerged. Usability testing focuses on how persons with a representative background evaluates a certain product acoording to their needs. You can read more about these terms in section 2 Theory.

When participating in school projects and/or during observations in usability testing we came to the conclusion that testing takes a lot of time. This may contribute to large costs for companies. These might be that they are not able to perform the desired and necessary number of tests or that no usability tests are performed at all. This can result in large costs.
During the test session much of the focus is placed on observation and documentation. We believe that information from the observations can be lost due to difficulties in attention when taking notes; when taking notes one usually focuses on the writing instead of observing. This might cause them to miss or misinterpret movement and errors during the test. Another experience is that even though one often videotapes the test

sessions in order to not miss anything, one usually does not use the material. The reason for not using the video material is probably due to the difficulty to know where to look, as well as the amount of video material to work through.

Therefore we wished to evaluate and develop an easy service that helped the testing personnel to improve the testing sessions and improve the ability to document and share the result. Part of the focus was to see whether the issues in usability testing could at least be partly solved by using a tablet.

1.2 Purpose and goal

The goal of this project was to develop a prototype that will simplify the handling of data during user tests and shorten the time needed for analyze it; as well as making it simpler to review the gathered data. A part of the goal was to find out how usability testing can be improved by using a tablet. The project aimed at providing a prototype with the following features:

1. Simplify the gathering of data during usability tests.
2. Simplify the analysis of information gathered from usability tests.
3. A useful tool designed for a large user group.
4. Create a compatible system for use on both a tablet and PC.

1.3 Limitations

Time has been the biggest restriction during the master thesis. Due to the limit of 20 weeks for both gathering of information as well as development and testing we had to make some developmental limitations. Therefore it was important for us too early in the project list the wanted features and weigh them according to importance in the final prototype and the time it took to implement them. The tool is developed for the Windows Surface RT tablet and programmed in C# and XAML. The reason behind the chosen platform is that the company, where the prototype was developed, had much knowledge regarding this specific platform.

The chosen platform and programming language was new to us at the beginning of the project and the lack of knowledge gave us some problems with implementation and optimisation which took more time than anticipated.

1.4 Target Audience

The tool is meant to be used in the industry or in academic research projects to collect observation and test data. It is primarily for the experienced usability testers, but it should also be useful for those with less experience. It should also work on smaller projects and for those who can not afford or use bigger and more advanced systems or just as an complement to the already existing test supplies.

1.5 Cenito Software AB

Cenito Software AB is a software development company located in Malmö, Sweden. With extensive expertise in development and design, they mostly create solutions for Windows based services. Their role in our master thesis project have been knowledge and guidance in the development.They pushed us in the right direction, gave us the "outside the box" comments and challenged us to go further. We were also encouraged to justify the design decisions and find the main parts within the solution. Cenito assigned us a supervisor for direct questions and project guidance and an office space for us to work from.

1.6 Division of labor

During the work on this thesis both authors have been working together in the same office at Cenito Software AB in Malmö. Naturally there have been discussions, proposals and teamwork during the process. The overall design of the application as well as the layout of the report have been done with mutual decisions from both authors.

Most functionality of the application has initially been created by one author, but then later on at some point been altered by the other author. But there are some areas where there is no doubt that one of the authors

have had more responsibility than the other.

A list showing the distribution of responsibilities between the authors is shown below:

Jessika

- Basic foundation of the observation template used during the test.
- The functionality to in advance prepare, edit and use a test template.
- The creation and editing of icons.

Sofie

- The functionality in the analyzation part of the application. The summation of total test data and the ability to connect it to a video file.
- Responsible for preparation and execution of the final test

Chapter 2

Theory

This chapter cover topics and areas of concept and knowledge that we have gathered during the feasibility study and as students at Lund University. Affected areas of the report should be seen as a quick and simplified guide to the terms and concepts that later gets brought up and discussed.

2.1 Usability

Usability is a term which can have many definitions, one definition is an ISO-standard, ISO 9241. This standard defines usability as "The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments." [1].

According to Schneiderman, one can ensure usability by creating an effective and well designed interface. In his book it says "Effective interfaces generate positive feeling of success, competence, mastery, and clarity in the user community. The users are not encumbered by the interface and can predict what will happen in response to each of their actions. When an interactive system is well designed, the interface almost disappears, enabling users to concentrate on their work, exploration, or pleasure." [2].

In a Handbok of Usability Testing [3], usability is defined as: "when a product or service is truly usable, *the user can do what he or she wants to do the way he or she expects to be able do it, without hindrance, hesitation, or questions.*". The writers also state "To be usable, a product or service

should be useful, efficient, effective, satisfying, learnable, and accessible." [3].

2.2 Usability Testing

The concept of usability testing is hard to define but it is often referred to as a process which uses persons with a representative background as participants in test to evaluate and validate how a product meets specific usability criterias. There are various procedures to how tests can be performed and what elements it should contain but a good approach is to start writing a test plan as soon as the discussions on future testing has begun. The test plan can later be refined and become more detailed as the project evolves.

The format of the test plan may vary depending on what type of test that will be performed. The test plan for the final test can be seen as an example in Appendix 8.4.3, but some of the following sections may be included in the report: [3]

- Purpose, goals and objectives of the test
- Participant characteristics
- Methods
- Test environment, equipment and logistics.
- Data to be collected and evaluation measures.

One of the more important tasks to conduct before the test takes place is to develop the test material needed to communicate with the participants and the data to collect, an example of data measurements can be seen in Appendix 8.3.1. This to get enough time to structure the documents and avoid future misunderstanings. The documents needed are often defined by the test plan and an example may be: orientation scripts, background questionnaire and task scenarios.
You also have to decide on a location and make sure that there are equipment available to record the pre-defined user data and whether there should be observers present. When the test has been performed and the data has been collected its time to go through and analyse the data according to the criteria described in the test plan and whether it is in line with the goal.[3]

2.3 User Centered Design

User centered design is a term that is often used in modern products and applications. The objective is to design useful products but it also focus at the users needs and centers the process around them. The aim is to support the methods that the user is already comfortable with without changing the user behavior. For the success of a good user design, it is good to divide the design process in various basic principles. [4]

Early focus on users and their tasks
Emphasis is placed on identifying the users and their everyday tasks. This should be achieved by a close work between users and designers to systematically and structurally collect information about and from the user.[3]

Evaluation and measurement of product usage
The interest lies in behaviour measurements and interaction with the product. Preferably, early in the design and development process use actual users to test the prototypes. By observing the interaction you can receive clues you would not get in a interview. Observartions make it possible to spot automatic interactions made by the user which would otherwise not be emphasised.[3]

Iterated design
To get the most out of the work, it is important to go through the results collected during the product's tests. By performing tests early in the design process and go through the results you get the possibility to early in the process change and rethink the design decisions.[3]

2.4 Donald Norman design principles

Donald Norman are an experienced worker in the field of cognitive science, design and usability engineering. To apply user centered design and structure the product these areas of interest should be applied.

Visibility
The placement of functions and details is very important for how well a user will notice the function and whether it will be used. By highlighting the important functions and/or divide the design in different levels. The design may lower the risk that the functions will not get "out of sight"

and result in users having difficulties in interaction with the product. By hiding irrelevant functionality and by bringing it forward when necessary you can lower the risks of misuse or misunderstanding.[5][6][7]

Feedback
Feedback focuses on informing the user that an event has occurred, the result of the action and lets them continue with the interaction. Different kinds of feedback are used in interaction design e.g. audio, tactile and haptic feedback.[5][6][7]

Affordance
High affordance gives the user a clue as how to interact with a function or product. As an example; a button provide the urge to be pressed, a handle wants to be pulled/held. When the affordance of a function or product are optimised it will be obvious to the user how the interaction should be made. [5][6][7]

Mapping
Mapping symbolises the relationship between a function and its impact on the environment. This can be represented by the play button on your dvd remote control. When play gets pressed the movie start running and you push the button with "two play icons" the frame runs at the double speed.[5][6][7]

Constraints
Constraints describes the functionality to restrict the user interaction to avoid navigation errors or misinterpretations within a certain functionally or the design. One example can be found in menu systems where unavailable information appears as faded to avoid interaction.[5][6][7]

Consistency
An interface with high consistency are defined by a design where similar tasks are presented in a similar way. This results in an interface where they use the same navigation method throughout the application, or that the menu is familiar and that the icons do not change the meaning between different pages.[5][6][7]

2.5 Interaction patterns and measurements

Microsoft published an article which describes different interaction patterns for Windows 8 and what to focus on when one strives to develop a good design.
Good design is not only what is available on the screen, it is also how well it has been adapted to the user. Another important aspects is to discover how the user interacts and holds the tablet, as this depends on the application.

Different people have favorite grips while using a tablet. The grips may however vary depending on environment, as well as duration of interaction. A designer has to have these aspects in mind when developing a product.
Figure 2.1 shows the areas which are focused on during interaction. Figure 2.2 demonstrates which areas a user focuses on while reading. While designing an application one must know how a user will spend its time using the application. Will it be used for reading or mostly for interaction.[24]

Figure 2.1: Areas focused at during interaction with a tablet in both landscape and portrait view. [24]

Figure 2.3 - 2.6 shows the most commonly used grips for interaction with a tablet. The different images each contribute to different areas where the information is available for the user and where is gets hidden by fingers and/or hands.

Figure 2.2: Areas focused at when using a tablet for reading, with landscape and portrait view.[24]

Figure 2.3: “One hand holding, one hand used for light to medium interaction with tablet.” [24]

Figure 2.4: “Two hands holding, thumbs used for light to medium interacting with tablet.” [24]

Figure 2.5: “The tablet is placed on table or legs, two hands used for light to heavy interaction.” [24]

Figure 2.6: “The tablet is placed on table or stand, with or without any interaction.“ [24]

Chapter 3

Method

Different methods were used during the project's development. These are shown below in the order they were performed.

3.1 Feasibility study

To gain insight in whether the proposed product was necessary and what its requirements would be; a feasibility study was performed. Below are explanations of the different methods used to conduct the study.

3.1.1 Literature

A literature study was performed by gathering information from different papers and publications regarding the subject of usability testing. The main focus was to find information regarding how tests were performed, which data was gathered and which problems that usually occurred. The literature was found by using Lund's University's search engine, LUBsearch and by following the references used in the relevant papers discovered by the search.

3.1.2 Interview

Interviews were conducted with the help of six participants with a background in both academic and industrial work. The participants performed their user testing in usability labs as well as observations in the work environment of the test subjects. To get an understanding of what

type of data they collected and which methods that were used we scheduled an interview. During the interviews we investigated the interest of the respondents regarding the development of a tool for usability testing and desired functions within the system. The interviews were of a structured sort, where we tried to avoid yes/no questions.

The questions can be seen in Appendix 8.1 and the main points of the questions were:

- What type of data are recorded and what methods are used.
- Difficulties with the used methods.
- Interest in the presented service
- Desired functionality of the service

3.1.3 Observation

An observation of a usability test was performed; which took place at the usability lab at IKDC in Lund. The purpose of the observation was to see the below listed methods and how they were used, and give us an insight into the problems and vulnerabilities that occur and how they can be prevented. We participated during two test sessions and afterwards we prepared some questions to ask the persons responsible for the test and hear their experiences from the usability tests. These questions can be found in Appendix 8.2. The main points of the questions were:

- What type of data are recorded and what methods are used.
- Difficulties with the used methods.
- Interest in the presented service
- Desired functionality of the service

3.1.4 Previous work in the field

The results from the literature study and the interviews presented examples of how usability testing is performed. With the urge to develop a new test tool we researched the field for available tools. These examples were then investigated by gathering information from the Internet, mainly from the tools' homepages.

3.2 Conceptual model

To get started with the project and defining our idea we chose different methods and strategies to get our minds ready and to be certain that we shared the same vision and goal with the prototype.

We started out with an individual brainstorming, we sat down separately to design and think about what features the application should contain and in what environments the product should be used. Afterwards the ideas were presented to our supervisors and the low fidelity prototype slowly started to take place in mind.

When the idea was planted and vision was clearer a group discussion was booked at Cenito with the CEO and the company supervisor to present our vision and through discussion decide the next move in the process. This was done to help us define the product further, and we were encouraged to write down user and product scenarios to make us see the wider picture and for the moment lose the thought on what it should contain, and rather what the user should do with it. Rather than it should have functions to measure time through different scenarios and have buttons to list specified functions and edit options. The scenarios turned into a storyboard and after a second group discussion we developed the final touches to the backlog and the low fidelity prototype and the code development could begin.

3.3 Software development

During the software development process most of the time was focused on discussions and continuous updates between us. We chose to divide the tasks between us to optimise the process and get the most out of the project. When we encounted obstacles we used brainstorming and group discussions to solve them as well as discovering different approaches.

The positive outcome from using an agile approach to programming while working at Cenito have been the knowledge of the personnel at the office. Not only the discussions with the project supervisor. The positive outcome of the discussions and tips during coffee breaks have given us feedback and guidance to different class libraries within C# and how problems have been solved in the past. A small library of books in app-development have freshed up the ideas when the inspiration was lost.

Since we used an agile and iterative approach it was very easy to develop and adjust the code as the project developed. During the feasibility study we got in contact with a interested focus group which we kept contact with during the process of low fidelity and high fidelity testing. The revisits have given us a sense of security in the application and provided us with many interesting views on what the product should contain and what areas to focus our attention to.

3.3.1 Low fidelity prototype

We created two quick hand-sketched prototypes for the main points of the system to be used during gathering of information. These were then combined into one final low fidelity prototype in PowerPoint see Figure 3.1, which had an alternative which was based on another handheld interaction see Figure 3.2. This alternative mostly had positions of different elements changed from the original. Another low fidelity sketch was created to visualise how the creation of new buttons in the testing template could look.

Testing

The test was performed on our project supervisors, on two separate occasions, by using the powerpoint sketches and present them as the Windows surface tablet. After the presentation the sketches were openly discussed to give us an idea on what areas to focus on next and make sure that we shared the same vision for the high fidelity prototype.

3.3.2 High fidelity prototype

After the testing of the low fidelity prototype was conducted, discussions were made together with the supervisor that there would be no further development or sketches on the low fidelity design. Recommendations were made that we should start to develop the draft of the product prototype. This to familiarise with the programming language and the programmes used to develop the application. Another point was to learn of the possibilities and limitations of the development environment we had chosen. It is possible to read a walkthrough of two of the versions of the prototype, in Chapter 5 and 6.

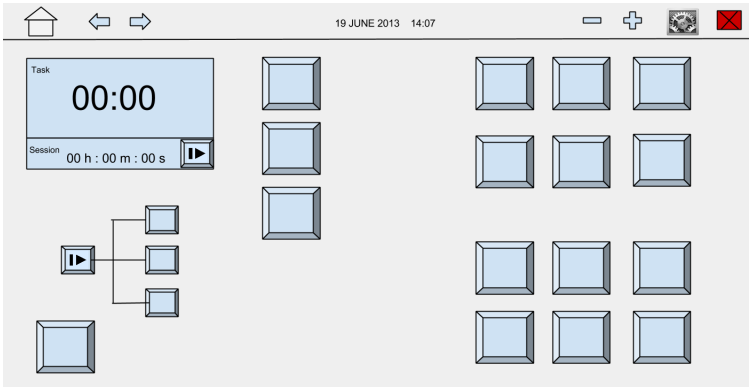

Figure 3.1: The PowerPoint low fidelity prototype for one-hand interaction.

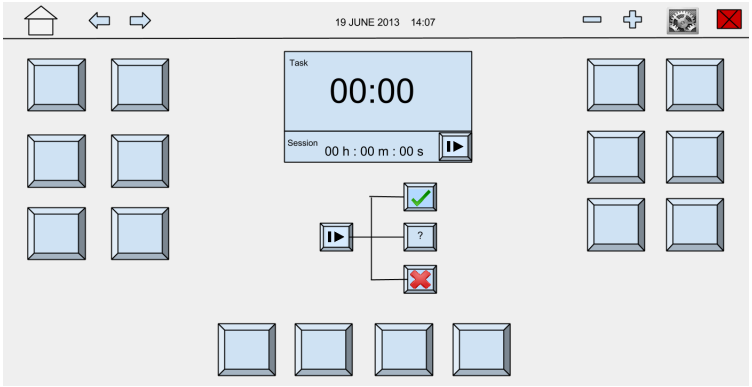

Figure 3.2: The PowerPoint low fidelity prototype for two-hand interaction.

Version 1.0: Features developed before the halftime testing:

- Two different layouts for the user to choose between, based on two holding positions.
- The ability to tag events and link them to the current time of action.
- The possibility to adjust the layout and create buttons according to the users specific request.
- The possibility to add a customised button during an ongoing test.
- After a finished test receive information about the test session, the different tasks and test result and the information about the tags pressed.

We proceeded with two models due to the fact that we could design them differently before the first test to try different approaches. This could differ in placement out of buttons, different groupings or a button that differs in name, colour and/or image.

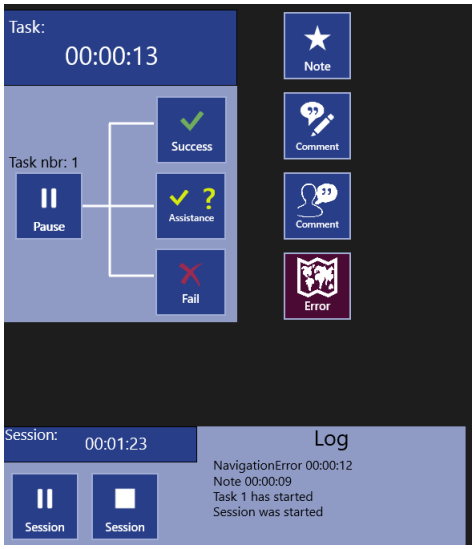

Figure 3.3: The updated user interface, with separated timers and a log.

Version 2.0: Features developed after the halftime testing:

- The task-timer and the session-timer was divided and placed in two different groups. This can be seen in figure 3.3.
- Scaled down to one version of the observer as the task-timer and the session-timer are now moveable.
- Added a log to the session-timer, which shows the actions performed during testing. This is shown in figure 3.3.
- Created a filter for the choices of icons, see Figure 4.10.
- Changed the interaction for choosing color, see Figure 4.11.
- After one has finished a test one can import a video and use the tags to navigate it.
- Divided the comment-button into two separate buttons. One for entering a comment in a panel and one for bookmarking a voiced comment, see Figure 4.12.
- Edited and adjusted the previous features for optimisation and further development.
- A function of saving both testing templates as well as sessions.

Version 2.1: Features developed after final testing:

- The panel for adding a comment, see Figure 4.13, now follows the placement of the enter comment-button.
- When using the keyboard one can use the enter-button as a confirmation.

Testing

Midway through the implementation a test was performed to make sure that the project prototype were heading in the correct direction. The participants were handpicked as an follow-up from the interviews that were held during the feasibility study. The test consisted of pre-defined scenarios where the participants had the opportunity to familiarise themselves with the application and get an introduction to the implemented functions. Afterwards an interview was held and they were asked to answer questions regarding the user experience and the implemented functions.

3.3.3 Code development

Development of the code has taken place in Visual Studio and Blend, two Microsoft owned developer tools. With the help of the two tools existing class libraries the application took form. To simplify the management of the code a revision control system was used, the system was TortoiseHg. To manage the development the coding have been prioritised with reference to the requirements specification listed in 2.4.1.

Testing

Iterative implementation has been used during the creation of the application; this resulted in small test sessions with the personnel at Cenito. The tests have been casual and with the goal to test a simple function or that a button has an appealing design and that the name and image accurately represents the task it should perform. This led to discussions and guidance in useful methods to simplify and solve minor bugs and/or coding problems. No official testing of the code have been performed, although checks and tests have been performed when the current functions have been developed and changed. When changes were made in the code it was directly tested to see whether it fulfilled its intended function.

3.3.4 Final testing

To validate the final product a final test was performed in the usability lab at Gambro, Lund. A test plan, see Appendix 8.4.3, was prepared and the final details were confirmed by our test supervisor at Gambro. The participants were handpicked from Gambro with the requirement that they should be familiar with the concept of usability testing and had some level of experience in participating or preparation of usability test. One of the four participants was ranked as an experienced user, due to having particpated during both our initial interviews as well as half-time test.

When the final test was booked we performed a validation test at Cenito to confirm that the test plan Appendix 8.4.3 and the test methods were developed and written in a way so that we would avoid misunderstandings during the test. Three employees participated in the test according to the pre scheduled methods and afterwards we received feedback and bug reports on the product interface.

The test consisted of four parts:

1. An *introduction* to the tool, followed by information about the test and instructions in how to edit a pre-defined test template. They edited the template according to what they wanted to measure by adding or remove tagg-buttons. This so they would feel comfortablewith the tablet during the test.

2. *Observe* a test being performed in the usability lab and with the tablet collect the data which they found relevant.

3. After the test they got to *analyse* the collected information and navigate between the different tasks and how the information was presented together with the recorded video information.

4. To summarise the test a short *debriefing* was performed afterwards and the participants answered questions on how they had experienced the test and the tablet interaction.

3.4 Tools

During the project different tools has been used:

- Software:
  - Visual Studio: A tool to develop code.
  - Blend: A complement to Visual Studio, where one can edit the code graphically.
  - Tortoise Hq: A tool to keep different versions of the code and to enable parallel development.
- Hardware:
  - Windows Surface RT: The windows tablet which the code is developed against and which the prototype has been tested on. It also has a removable keyboard, which has sometimes been used, especially in the preparation stage of the final testing.

Chapter 4

Result

4.1 Feasibility study

4.1.1 Literature

There were many studies and comments regarding the cost and time consuming usability studies and how one should be more efficient in use of metrics and in which tools to use. A solution presented in [8] was to try to automate testing. Other studies recommended which metrics had greatest impact or how to simplify the testing. Therefore; the main goals of the different studies indicated there is a need for greater efficiency in usability studies, both in what metrics are gathered as well as in the analysis.

Data gathered

The literature study showed that two types of information is gathered during usability tests, performance and preference data. It was also discovered that different studies gathered different information of these two types, depending on what the studies main goals were and depending on the limitations of the tools used in the studies.

The data which is gathered can greatly differ between studies in the quantity of metrics and also in whether the focus is on performance or preference data, which shows the width of usability metrics. However, the main focus seems to be on metrics which gather information regarding

efficiency, effectivity and satisfaction, and this data is mostly gathered by metrics of task completion, task time and comments and/or questionnaires.

Below are examples of usability attributes and their metrics, divided into performance and preference data, which different studies mentioned in relation to usability studies.

Performance data

Below are the main points of performance data which are gathered, for more information regarding specific metrics, please read Appendix 8.3.1.

- Comprehensibility (readability) [9]
- Effectiveness [10] [9] [11]
- Efficiency [10] [9] [11] [12]
- Error [10] [9] [13]
- Learnability [10] [9] [11]
- Learning performance [10] [9]
- Memorability [10] [9]
- Simplicity (complexity) [9]

Preference data

Preference is gathered by gathering information regarding the test subjects satisfaction [10] [9] [11], this might be done by questionnaires [14] [10] [11] and comments [14] for example. Further examples are in Appendix 8.3.2.

4.1.2 Interviews

Among the participants the tests were conducted mostly in usability labs, some of them also observed their subjects in their own environment to observe user patterns and/or test the product in real time scenarios. Tools used to collect data consisted of video and audio recording devices and pads for taking notes.
Most of the participants also said that they use structured interviews or

questionnaires to collect background information from the subjects. During the tests most of the participants used scenarios to let their subjects interact and perform tasks within a predefined pattern. During observations or test in the real time scenarios the tasks are already defined from the day to day activities and can result in difficulties with interviews or the ability to ask questions.
After the completion of the test the test leader often do a second interview or let the subjects answer a survey and discuss the procedure and reflections from the test.

The data recorded from the tests consisted of the time it took for the subjects to perform different tasks or scenarios, fault handling, errors made and how often they appear and information about the user experience.
How did the preference and performance aspect of data differ between the participants? One of the representatives from the industrial section has stopped measuring time during their usability test. This was due to the fact that their testing is almost entirely done on subject novice based knowledge with the prototypes. If you do a second test or with a subject that has another level of knowledge it can be interesting with measurement of time. They also mentioned in the interviews that they record video-data during the test sessions but the information is not reviewed afterwards, "there is not enough time to analyse the video-data". Another representative from the academical section does not take notes during the test session, notes are later taken during the analysis of the data. The participant dit not want to make preconceived conclusions from one single action or incident.

The overall interpretation from the interviews is that performance data is prefered in the academic sector with video-data and notes from the interaction, and the preference data is more prefered in the industrial sector with a larger interest in the number outcome from questionnaires and measurement scales. But they all agreed that the most important thing is to do a proper pre-study and analyse where the importance lies and what the tests aim to prove.

From the interviews we retrieved information about tools they used to analyse their data.After presenting our idea of an analysing tool to them we received feedback of what functions they saw an interest in. Opinions on the final product was that less is more. It was also mentioned that the tool should have one to three pre-defined functions and maybe four to five

functions the test leader could choose and specify for the different test situations. Tags or functions they would like to use and measure were, start and end of an task, emotions from the subject and what types of fault that occurred during the session.

There was also a large interest from most of the representatives that they could group the recorded data depending on task and/or product. The possibility to extract data from the tablet and import it to a stationary computer was desired for further analysis. One function desired and commented from two of the participants were the function to be able to communicate with the test leader as an observer. In situations when recording devices misses or have difficulties to record audio or video the observers would like to inform the test leader without disrupting the interaction during the test session. A chat function that does not distract the test session was one possibility that was discussed. Products on the market that did some or most of the desired functions are NVivo and Noldus the Observer XT.

4.1.3 Observation

During the observation the testers gathered information through background surveys and by videotaping the session. They focused mainly on the comments the participants made. Another focus was to discover bugs and they also wanted feedback regarding whether different parts worked smoothly. This was due to the fact that they were mainly trying to validate the product as it was in its final stages.

When answering our questions they let us know that they would have appreciated a tool which could quickly present them with statistics as this would speed up the analysation of the results. The data they would like to have gathered was mistakes in the different tasks and common comments, as well as audio/video. Another requirement was the function of sorting the gathered data depending on age, gender and so forth.

Regarding using a tablet as a tool, they also required that they would not need to place too much focus on the screen and that they would be able to use the codes they had constructed for common types of mistakes/comments.

During the observation, one of the comments from the interviews was

confirmed. This was the comment that observers might want to communicate with the test leader during tests, but this is difficult without revealing oneself to the participant. This confirmed the need for communications between test leader and observers.

4.1.4 Previous work in the field

This section covers references to similar products available today.

Data analysis software

The software aims to help users to organise and analyse data for users with different background and from different sectors. The software are open to different kind of methodologies which lets the user specify its own workspace and it handles most data types from text files, video and audio.

The user is involved in the projects and can create its own coding formats to help group the data into different patterns so you later can see how frequently different topics or content are mentioned in the data and automatically let the software link the data to different sources and see how they are associated and find undercover connections. The data and result can later be exported to other programs to share with colleagues and friends. Programs using this method is Noldus the Observer XT and NVivo.

The Observer XT developed by Noldus Information Technology is also available as an pocket observer for easy access and designed for field observations. The portable software can be administered on a PC for further use on a Windows or Android phone.

NVivo is a software developed by QSR International is specified for the Windows operating system and the program is designed after Microsoft user interface guidelines to give the user a familiar feeling. [15] [16] [17] [18]

UTE is a tool which evaluates and analyses data from websites and web applications. The tool method is quantitative and the data collected and analysed is manly measurable data. Morae is a software developed for tracing user data from audio- and video data, on screen activity and input from keyboard and mouse clicks. Together UTE and Morae can

be combined and return the same functionality as the programs listed above. [19] [20]

Web based services

One method available today is a remote web based service for usability test. The tests are outsourced to individuals where they perform tasks and tests at home with their own instruments.
The participants are required to have a computer and a webcamera where the individual can tape themselves while using the product. They also use TA, talk aloud, to describe the product being tested with its advantages and disadvantages.
This service gives the developer an easy process without any demands on access to usability labs and money can be saved. However this service does not give the developers any contact with the individuals connected to the service, they have none or little insight in the recruitment of test persons. If problems arise with the results the developer has no possibility to contact the participants for further questions about the interactions made. [21] [22] [23]

4.1.5 List of functions

Below is the functions required to fulfill the concept of the system, with the main goal to minimise the need to take notes during testing.

Required function

- To be able to flag events during usability testing.
- Synchronisation between flagged events and video.
- Be able to adjust which events to flag.
- Video viewing; to be able to view video of tests and to choose section to watch by event or task.

Desireable functions

- Adjustment of flagged events after testing.
- Be able to enter video from observations which are not performed live.

- Receive statistics from the performed usability test.
- Support of interviews and background forms.

4.2 Conceptual design

The concept is to produce a program which would increase the efficiency of usability testing; which would be done by decreasing the time and focus needed for gathering of data as well as the pre-work for analysation. This will hopefully give the user more time to focus on the analysation of the data and/or time savings overall. Focus is to try to minimise the amount of notes taken, which the analyst would have to go through in analysation and to automise some creation of performance data.

A main point is that the program should be independent regarding which environment it is used in. It should be portable and be functional in both usability tests as well as during observations/tests in the test subjects workplace. Another point is that one should easily be able to adjust the program depending on what will be tested and in what environment. The goal is to cover basic needs when testing usability or performing observations and that the program will replace and/or combine different aids which have been used earlier. The program will make it possible to perform usability tests without the aid of observers to gather data; as one can perform the test as well as gather data at the same time.

4.2.1 System

The system should support Windows 8 and run on a windows tablet. It is desirable if the system can be synchronised with a PC for further work and analysis of the collected data. Also, the data should be exportable, to for example Excel.

Regarding interaction styles; we mainly focused on direct manipulation [2] with an influence of menu selection [2]. One of the drawbacks with direct manipulation is that the users are required to learn the meaning of the visual representation [2]. The drawback will be remedied by using universal icons and by letting the users themselves manipulate the layout and buttons of the system.
There is no need for natural language [2] due to non-existent need to learn new syntax as the system's users will be experts in the area of usability.

4.2.2 Mental models and metaphors

The system uses the mental model of a media player and shows the corresponding timeline; which the users should be allowed to use freely. One difference from a regular media player is that instead of having the timeline separated by chapters, the system has separation by task and bookmarks of certain events.

Different metaphors are used for the system, one of them being that the system is a notebook where one can gather all data quickly and later retrieve it. The flagging of events in a film will have the metaphor of bookmarking. As one can bookmark a website, a place in a book and so forth, one can bookmark a part of a video with different kinds of bookmarks. Another metaphor will be used for buttons, such as using smileys for icons when indicating certain feelings expressed during the testing process.

4.2.3 Requirement specification /Backlog

The discussions during the conceptual model resulted in a specified number of functionality we wanted to apply to the application.

During test
The functionality of the application should be listed as seen below during an interaction:
Button the buttons on the screen should when pressed:

- Register the interaction that was made.

Clock the clock on the screen should be able to:

- Start, pause and stop the time when requested by the user.
- Register the time when a button interaction has been made.

When the functionality was implemented and working we wished to further implement functions to make sure the interactions still worked flawlessly and according to the desired functionality.

After test
When the test session is over the following functions should represent and/or represent the interactions made:

Timeline, there should be a visual timeline that shows the tag images of the interactions made during the test session.
Video, the recorded video data from the test session should be compatible with the button interactions. So when watching a timeline the user can depending on the tag-events navigate in the video clip.
Filter, it should be possible for the user when analysing the data to filter the test data according to a specific task or button interaction.
Data collection, after the test has ended the user retrieves information about the test which we call statistics. The user should then be able to retrieve and view:

- The button interactions made during the session.
- Time, both when a interaction have been made and how long it took for a task to be completed.
- Be able to save data and export it to a e.g. Microsoft Excel sheet.

Prior to test
The functionality of the application should be listed as followed before an interaction: *Adjust buttons*, the user should have the possibility to adjust:

- The button text
- The buttons background colour
- The icon of the button.
- Position, the user should be able to choose where the buttons should be placed to ease the interaction during the test.

The user should also be able to adjust the layout, ie. decide the position of all items in the testing template. This would create a better mental model as the user would be able to recreate their own model instead of the developers perceived image of it.

4.3 Development

This section describes the development and design choices that have characterized the different versions and how the test results have mirrored the continued developments.

4.3.1 Low fidelity prototype

The development of the low fidelity prototype consisted of both paper prototypes and redesigned sketches in power point. The design choices where based and described after Donald Normans design guidelines.
Visibility
According to the similarity between other applications and webpages the placing of the menu are located to the top of the page. The menu includes functionality for editing the page and navigate between them.
The icons for navigation are symbolised of a house and arrows in the left and right direction equal to the Windows Internet Explorer Web Browser. The red cross are also a function inspired from the web browser to push when you want to exit the browser. The buttons for editing the testpage are symbolised as + and - to give the user a feeling that they can add or lose information on the site.
Feedback
No feedback were used to improve the user experience in the low fidelity prototype. Although implementations were made so that the session/task clock would update itself during the test session.
Constraints
The play buttons have the functionality to start the session and task. The button also includes a function to pause the session/task. This makes is impossible to operate the timer (pause) when when the required function has not started.
Mapping
Buttons with functionality that affects the time of the session is listed around time window. The task buttons are also placed near the task window and to ease the use of the start and stop functionality to the task they are linked together with a black line to improve the navigation between the buttons.
Consistency
Buttons that affect the test session has the same design as a key block from a keyboard. These to differ them from the navigation buttons listed in the top menu.
Affordance
The function for listing tags are designed as a keyboards key, this gives the user the feeling that there is a real button on the touch surface and leads to correct interaction.

The transition between paper prototypes and the ones made in Power-

point made us rethink some of the design options. To simplify the design and avoid further confusion with test persons we chose to keep most of the buttons clear. With no picture och text at the button it can be used for any type of interaction and rather let the testperson come with ideas of what is should be and symbolise.

4.3.2 High fidelity prototype

Version 1:

The following functions were implemented in the first version:
Observer: This class is developed to contain functionality in how to document the usability tests and observations. It contains functions to:

- *Recording time*, both the time spent during a specific task and the total time of the session.
- *Buttons* to represent different actions made during the test session to help the collection of user data.

Edit: Its hard and almost impossible to develop a product which registers all possible user interactions during the test. Therefore implementations were made so that the user can create his/hers own buttons. There are two possibilities to adjust the buttons on the interface:

- *Quick add*, during the test session the user can create a button either by choosing the default button or design as desired.
- *Edit mode*, before the test session takes place the user have the possibility to add new buttons or edit the buttons already placed on the user screen. Either by using the default button or design as desired. When chosen the wanted button the user can place it wherever they want it to be positioned on the screen.

Statistic: This class retrieves information on collected button interactions. This to give the user a feeling on how the interaction has been performed. The information listed are:

- The buttons pressed during a specific task and when according to the start time it was pressed.
- How well the task has been performed depending on which button the user choose to press when finishing the task.

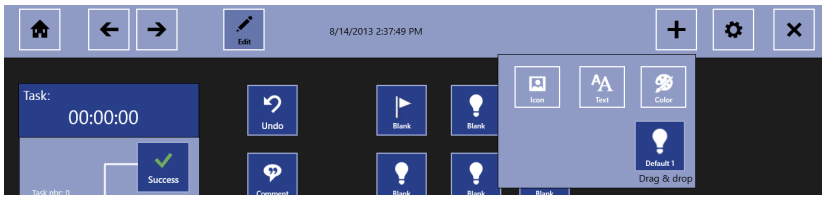

Figure 4.1: The menu for one-held version of the system and the add-button panel is enabled.

Figure 4.2: The menu for one-held version of the system and the add-button panel is enabled.

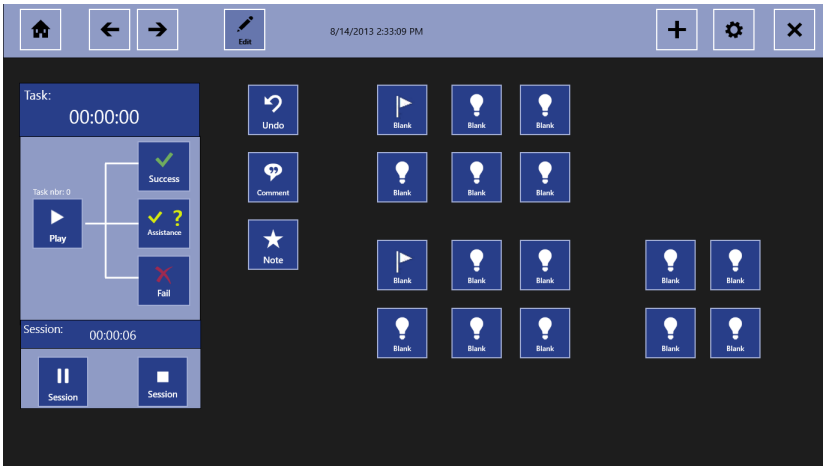

Figure 4.3: The observer adjusted for one-hand interaction.

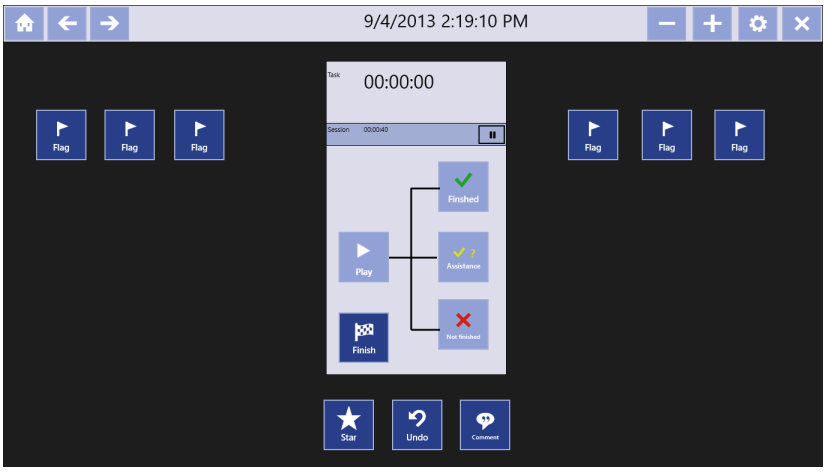

Figure 4.4: The observer adjusted for two-hand interaction.

Figure 4.5: The statistics mode of the prototype version 1.

Design choices:

Layout and grid - The observation class was developed into two versions, designed according to the guidelines listed in figure 2.3- 2.6, one with focus on one-hand and one with two-hand interaction.

Top menu: Minor actions distinguishes the two models to each other. The menu is developed for right handed interactions where the buttons are placed according to their importance and how often they are used during the session as seen in figure 4.1 and in figure 4.2. Navigation to the left according to a one time interaction and a forced and well thought interaction. Editing to the right, these interactions should be easy and fast to perform. When a button is added during the session the user has none or little time to make the change needed. This resulted in a right alignment with a quick default button to position on the background. The default button is shown in figure 4.1

One-hand: With an interaction technique where the user navigates with one hand or even from a distance with the tablet placed upright the focus area was sectioned to the right with a majority of right handed users. This resulted in a button interaction area placed to the right where the buttons for tagging events was placed for easy access and the timer to the right to restraint the user and reduce error clicking where the hand mistakenly touches the buttons which pauses or stops the test session. The resulting interface is shown in figure 4.3

Two-hand: With an interaction technique where the user mostly navigates with their thumbs on the screen the most important information have to be placed near the corners of the screen and the information needed for a smaller amount of interaction or only for observation centered on the screen. This resulted in two button interaction areas placed on each side for easy access. And the timer menu centered on the screen made it easy to observe. The interactions made to the timers is made after an active choice made by the user and does not occur as often as the interaction with tagg-buttons. This results in an active choice by the user and reduces error clicks where the hand blocks or touches the timer buttons by mistake. The resulting interface is shown in figure 4.4

Statistic: The statistic view is meant for session feedback and reflection. Therefore the text window and timeline have been centered on the screen, see figure 4.5 according to the guidelines in 2.2. This highlights the information on the screen and gives comfort when observing the session information and navigating between the tasks performed.

Sizes and measurements- According to the Windows touch interaction guidelines the buttons were required to be at least 7x7 mm or 40x40

pixels, as this is the recommended minimum size.[24]. Since accuracy matters in the prototype, the size should be 9x9 mm, 50 pixels at least. The size of the tags ended up being 90 pixels as the recommended 9 mm was confused with pixels. This also ensures that the error rate is kept small [24]. According to the guidelines in [24], there should always be a minimum of 2 mm, 10 pixels, in between buttons.
Colour - The colours chosen for the applications were different shades of blue placed on the "typically" Microsoft black background. The blue color is neutral and attracts a wide number of users. The shading divides the different areas e.g. task related buttons from the ones used in the session area and the shades are also used in the top menu to differ them from the buttons designed for tagging events and maneuver the timer. A more contrasting purple is used when error messages or confirmation popups are shown. Otherwise it is possible for the user to edit the background colours of their "tags".
Icons - The images in the top menu and on the buttons were mostly collected from the "Segoe UI Symbol" list [25], [26].The images are designed with a minimum amount of details to give a clear idea on what they describe to make the application look tidy and manageable.
Typography - The prototype uses the default typography for developing of Windows apps.
Sound - With a product developed for usability testing and with a majority of time spent in the same room with a client you do not want a device that reminds him/her of the interactions or errors made during the session. Therefore no sound or haptic feedback were used for the application.

The design choices were based and described after Donald Normans design guidelines [5], and they follow the modifications made from the sketches described in the low fidelity prototype.

Visibility
No major changes have been made to the visibility between the low fidelity and high fidelity version.
Feedback
Button pressed - When a button is pressed its background changes colour for three seconds to illustrate that a interaction has been made.
Button position - When a user want to place a new custom button on the screen a pop-up window appears after the button has been successfully positioned. The pop-up window asks the user if the position is correct

and depending on the answer the user can further move the button until it wants to lock the position.
Edit - In edit mode the buttons available for interaction turns darker in the menu field to illustrate the possibility to add buttons and save the created result.
Copy - When re-editing a button in the edit mode, the user retrieves a copy of the button to visualise the button which should be redesigned. The changes are made to the button and the user can in realtime see the changes being made and see how the result will look when accepting the new design.
Constraints
The play buttons which start the session and task also have the functionality to pause the session/task. This is implemented so when the user pushes the play button the appearance changes and take the form of a pause button.
Mapping
Buttons with functionality which affects the time of the session is listed around time window. The task buttons are also placed near the task window and to ease the use of the start and stop functionality to the task they are linked together with a black line to improve the navigation between the buttons.
The three buttons to choose between when finishing a task have been designed as a metaphor for traffic lights. With the colours green, yellow and red we want to give the user a quick introduction into how they should be used. *Consistency*
All the buttons are designed the same way. They consist of a frame with an image and a text inside, this to guide the user on what interaction should be made and what the result would be.
Affordance
All the buttons have been designed with a frame around it. This to make them pop out from the background and give the user a metaphor similar to button interfaces and give them a clue that they can be interacted with.

Test results:

The testing presented varied results, both successful and less so, with common and more uncommon issues and comments.

Task/session, startup and feedback

Most of the participants experienced problems at the beginning of the test, where they felt that some part was missing and resulted in some confusion regarding whether a task and/or session was running. Some

also had problem with distinguishing the session from the task specific button and what their purpose was; they felt that this part could be more clear. A part of the confusion was confirmed to have been caused by language misunderstanding as the test was performed with a Swedish scenario-list, Appendix 8.4.1, and English was used in the prototype.

All but one participant expressed a desire for more feedback when using the prototype, the color change was good but they wanted something more to really know that a button had been pressed. Different ideas were to use a timeline, a log of actions but also a checklist of prepared tasks/scenarios so one knows where in the testing procedure one is. None of the participants voiced any need for haptic or audio feedback, and half expressed that they strongly did not want audio feedback as this could disrupt the testing. Two also agreed that haptic feedback could also be a disturbance during testing, due to the sound vibration makes.

Some of the participants also felt that some steps were missing, such as a more suitable start menu and the step to enter information regarding the test prior to starting the testing session.

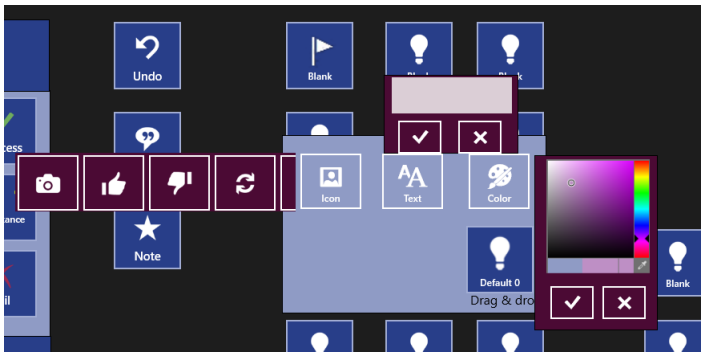

Figure 4.6: When all editing options are open.

Interaction issues
There were some issues regarding interaction in general, such as the non-disappearance of the blue panel and the activated options when one tapped outside of the blue panel in Figure 4.6 during creation/editing of a button. One participant also wanted the activated options to disappear when one clicked in the blue panel and another wished for a cancel-button

when adding new tag-buttons.

A discovery was that all of the participants chose to singletap when selecting the icon for a button instead of double-tapping. None discovered the double-tap without help and all felt that they had selected an icon using only single-tap. Therefore, they expected the icon-options to disappear after a single-tap. Another discovery was that most did not discover that repeating a tap on an optionbutton, such as the add-, Icon-, Text- and Color-button seen in Figure 4.6, would yet again hide the option. There were also questions regarding when they wanted to move buttons and which ones should be movable, however only one wanted all tag-buttons to be moveable at all times. Others wanted only to be able to move buttons while editing the test template.

During testing it was discovered that some of the participants wished to remove buttons they had added during the session, especially the ones which had been done incorrectly. While creating/editing buttons, a third of the participants experienced problems with the scrollbar of the icon-options as they did not perceive it to be scrollable; however, the others did not encounter this problem. Some of the participants mentioned a desire for the possibility for categorising and filtering the available icons as well as having the icons prioritised after usage.

Certain buttons

- Save: Half of the participants were confused by the save button, either regarding what its purpose was or that they thought it should not be needed. They mostly felt that the testing template they had created should be saved automatically and without the need of a save button.
- Comment: All of the participants thought that they would be able to enter information regarding an action when tapping the Comment Button and not only marking the event of a comment. They all requested a way to either enter information directly, document on paper or at least a way to enter information after the simulated test was done. Some were positive to a numbered comment-function where one could number the comments made on paper to know where in the test they fit in. Half were outspokenly positive to the use of an Anoto-pen to be able to get the comments on paper into the system and saved together with the rest of the gathered data.

- Undo: Most of the participants were very unsure of what this tag-button did, some thought it would undo what they had entered while others thought of it as a mark of when the simulated test persons would perform an undo action during testing. The ones who thought it was used to undo their own action did not want to use it as they were unsure when and what it would undo. Most felt that it should only reflect the test person and not themselves and that they should instead be able to adjust the tags after the testing session.

  Minor issues:

  - Language: There was some problem with the use of language, especially when translating from swedish to english. Some difficulties were due to the test persons experience and own usage of terms.
  - "Copy": The wording when showing a copy of the button chosen to edit in edit-mode was wrong, as one participant thought this meant it was a copy and would result in a new button. A better wording would be "Preview".
  - Default buttons: A third of the participants wished for the possibility of default buttons, such as buttons for bugs etc. A bonus would be if these were sorted after how likely they were to be used.
  - Unclear popup: Some thought that they would be able to move their newly created tag-button after accepting the position in popup box. This was in one case due to not reading it. The transparency of the popup also caused problem for one participant due to problems with eyesight.

  Confirmations:
  It was confirmed that two thirds of the participants preferred the flag icon compared to the lightbulb as a default tag icon, only a sixth did not like either and preferred something else.

Most also liked the drag and drop actions, even though there were some minor misunderstandings in the beginning. Three of the participants first tried to use two fingers to start editing a tag, one holding the tag in question and one trying to tap the edit button positioned at the top menu. Two out of these three discovered

by themselves quickly how to perform drag and drop, and where not even sure why they had tried differently from the beginning. Most participant mentioned the mental model of mobile phone usage when interacting with the tag in connection to editing and removing.

Many of the participants voiced that the prototype was easy to understand and had good learnability. They also thought that they would most probably know how to operate the prototype without any problems the next time.
The color choices were confirmed to be okay, there were however some issues with the color choices in Observer 2, as some buttons were perceived to not be available due to their lighter color. Half of the participants also confirmed that they prefered Observer 1 over Observer 2, while only one participant preferred Observer 2. This meant that most used a one hand grip to handle the tablet instead of the two hand grip.
They also confirmed that they would like to be able to compare the results from different testing sessions with each other, such as specific tasks performed by many participants.

Comments and ideas:
The participants gave some comments and ideas; which were interesting:

- The need for a "Help" tag, where one could indicate where the participant received help.
- Tags with different gradation of a tag, for example tags ranging from minor to major bugs.
- A tag which says "nothing", which one could use as to not give away when collecting data to the participant. This as a way to always be able to press a button.
- The possibility to prepare a test session by adding information regarding the tasks to be performed, such as what tasks they involve and time limits.
- A button to skip tasks when time is dire or when previous results makes a task redundant.
- One participant thought that by double-tapping a tag one would regret the action.

- The idea to use folders, such as in iPhone, to gather similar tags and minimise clutter.

Bugs:
Some bugs were discovered during testing, these were:

- New buttons and/or the copy of an edited button got stuck in the front of the prototype, for example before the add/edit panel.
- Certain icon choices were missing in the scrollpanel for icons.
- Issues popped up when the popup for accepting buttons position was declined without following up with a confirmed position. Specifically if the decline was followed up by trying to add or edit a new tag-button.
- Vertical holding was enabled which resulted in a scrambled UI where certain buttons and options were not available.
- Neither the virtual or the real keyboards enter button worked as an approval for the new text when creating/editing a tag-button.

Resulting actions:
The test resulted in a new backlog of actions to be done or reflected over, which were:

- Clarify / separate session and task actions and information.
- Decrease the transparency of the popup for accepting a new tags position, seen in Figure 5.6.
- Feedback: Create a log or timeline to give the user more feedback.
- Create a start menu, to create a better flow and a template.
- Edit the edit function:
    * Create a cancel button for adding tag during session.
    * Visibility of options, what shown when.
    * Colors: enable the choice of more than one color.
    * Reflection of when a tag should be moveable and/or removable?
    * The scrollbar of icons:
        · Find a better way to show that the bar is scrollable

- Enable a filter by having categories, such as favorites and prioritised icons.
- Change the interaction when choosing colour, to simplify use.

– Comment button: Decide what its purpose is; options are: to write your own comment, using numbers to clarify which comment fits where, or quick choices of pre-defined options of comments.

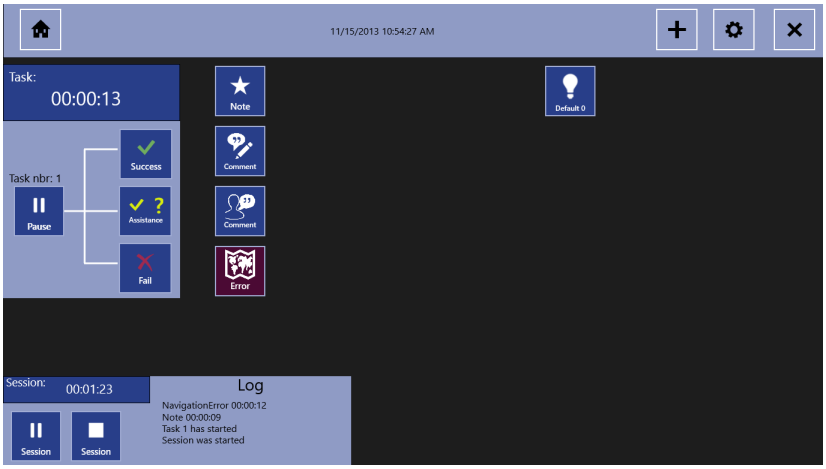

Figure 4.7: The updated interface of the observer template.

Version 2:

The result from the half-time testing made us perform some changes. The updated interface of the observer template is shown in Figure 4.7 and below is more details regarding other changes.

Design choices:

Constraints:

Constraints were added, these were the checks whether one wanted to exit a part of the program with or without saving, as well as a popup which showed prior to starting a test. The popup enabled the user to exit the testing session or to start it.

Layout and grid - In this version we did not make many changes to the layout, the most apparent change is the separation of the timerpanel into one tasktimer-panel and one sessiontimer-panel with a log. A grid was created to simplify the placements of objects and make it easier to place objects in straight lines.

One-hand: This interaction method was the new focus for the prototype as all participants used this method; and all but one prefered this way.
Two-hand: Was discarded as the one-hand interaction was prefered by the users during testing. Furthermore, there was no longer a need for separate interfaces as all objects were going to be movable, which would make the interface adjustable for both interactions.
*Statistic:*The following changes were made in the design, also see Figure 4.8:

- Timeline - The timeline has been extended and it also draws vertical lines symbolising the tags collected by the tablet.
- Comboboxes - An extra combobox was added to list the tags pressed, as to ease the analysis of the collected data. The fontsize and width of the previous box have been changed to improve the interaction.
- Textboxes - Informative text boxes have been placed over the comboboxes to inform the user of the information listed.
- Export - A button added to save the collected data.

Video: This is a new function for this version of the application. The GUI is reused from the statistic mode to strive for consistency between the two analysation modes and can be seen in Figure 4.9. The differences in the design are the following:

- Combobox - One difference in this mode is the handling of the tags, instead of grouping them as in the statistic mode the tag receive a unique ID to separate them. The ID was based on what order the tag was interacted with according to the session. e.g. 1. Comment. This in order to get the correct location for the "right" tag in the imported video. An extra combobox to handle the video offset have been added. This to give the user the ability to change the position in the movie where the tag is represented.

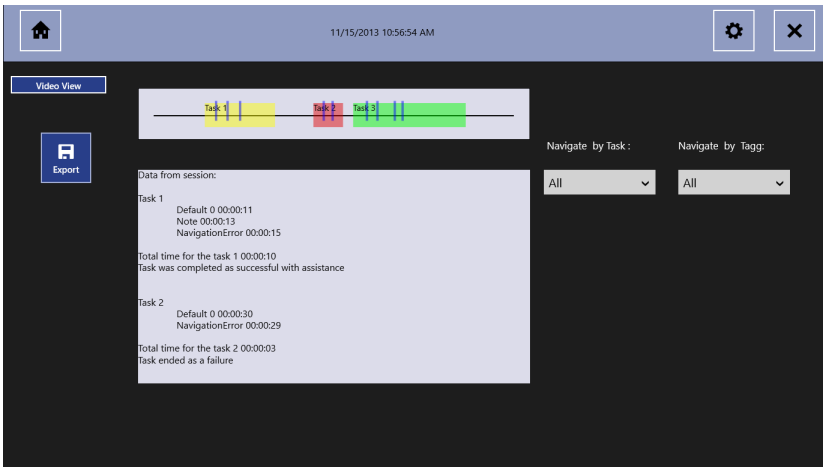

Figure 4.8: The statistic view in version 2 of the prototype.

Figure 4.9: The video view in version 2 of the prototype.

- Import - A button to import a mp4 video file.
- Mediaelement - Instead of a textbox a mediaelement was added to play the video file which allows the interaction between a tag and a scene in the movie. Play and stop buttons to handle the video have been placed under the mediaelement.

Edit: Changes were made to interaction of adding/editing a tag, one can now view more icons and filter them see Figure 4.10. One can also change the fontsize of the tag's text as well as a new way to change the color of said tag, see Figure 4.11.
Changes to buttons: Changes were made to some buttons/tags.

- Comment-button: This tag was divided into two different tags, see Figure 4.12. One for entering a comment and one for marking the event of a voiced comment, see Figure 4.13.
- The undo-button was removed and a navigation-error tag was added to the default-buttons, see Figure 4.7.

Sizes and measurements - Stayed the same.
Colour - Stayed the same.
Icons - Icons were added, mostly from the same source as earlier, the "Segoe UI Symbol" list [25]. Other images were downloaded from [27] and some of these were edited by us to create the icons we wanted, see Appendix 8.5. A few icons were created from scratch, these are shown in Appendix 8.5.
Typography - Stayed the same.
Sound - Stayed the same.

Test results

Observations:
While in the testing phase of the test, the participants worked in varied ways. Different approaches were to gather every move the test's user made navigation wise, while another only gathered information sparingly with focus on comments. The user who had previous experience with the prototype worked very calmly and methodically.

Interaction issues
Almost all of the participants experienced problems with how to

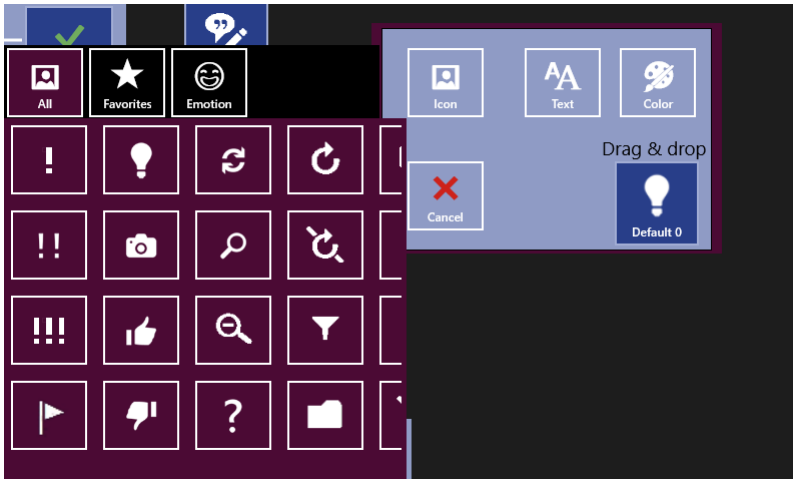

Figure 4.10: The filter menu to filter icons.

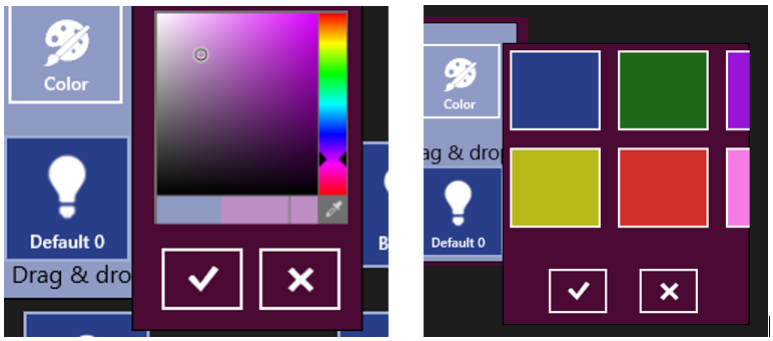

Figure 4.11: The choosing of color has changed from left image to the right image.

Figure 4.12: The two comment buttons, one for entering and one for marking comments.

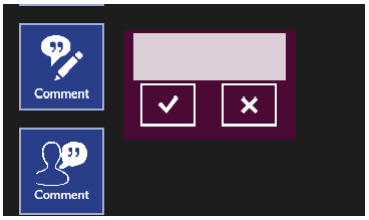

Figure 4.13: The enter comment-button with its panel enabled.

enable editing of buttons, as they did not first see the options appear in the menu and when they did they were not always sure at first how to enable them. One user tried to use the add button and two others tried to use two fingers. In the end all of them realised how to enable editing, but some asked for another way to enable it or that the options should be made more visible. One requested that the options should always be visible while another wished for a popup.

Another interaction issue was that one of the users tried to perform drag and drop on the preview button for an edited button. A quarter of the participants did not see/use the filter for the choosing of icons, some expressed that it was not very visible and did not stick out from the choices of icons.

Issues:

- Big:
  - No way to regret taps on buttons in testing mode, this caused annoyance and frustration.
- Small:
  - Finding the export-button in statisticview.
  - The need for explanation regarding the difference between the two comment-buttons.

Comments and ideas:
The participants gave some comments and ideas; which were interesting and confirmed earlier ideas:

- Half of the participants wanted to be able to name tasks.
- An idea was to be able to connect specific tags to an input-panel, so as to be able to enter comments regarding the reason for tagging. For example if the tag was "Serious mistake", it would be great if one could directly enter what the serious mistake was.
- Another idea regarding comments was that they should be viewable in the video, this was expressed by two participants. One of them also wanted to be able to add and edit comments while watching the video.

    - A follow up idea was that one of the users wanted a button for quick navigation between tags in the movie, a "next tag" function or even better a "next comment" function. If one had already chosen one kind of tag one should easily be able to navigate using a next function for said tag.
- A comment regarding comments was that instead of displaying in the listview what kind of comment was entered, it should display the comment. This was expressed by half of the participants.
- Two participants wanted icons to be shown in listview, as this would give them an easier overview. One also expressed the wish to list only the fault-tags; specifically all the tags involved with registering faulty behaviour.
- One participant wanted to be able to use the tablet for taking snapshots during testing.
- The need for volume buttons in video mode was expressed by one participant, so one could be able to watch the video with and without sound depending on the situation.
- Half of the participants wanted to have a timeline only for the video, where one could see where one was in the video and how much of it was left. One also specifically voiced the want to use it to scroll freely in the video.
- The same half also wanted to be able to watch multiple tests at the same time. However, they did agree that this should probably be done by a workstation and not on a tablet.
- One user wished for a checklist, where one could have the different steps required for the performance of the task. These steps would then be checked or not depending on failure.
- The function to export the data to excel was mentioned by most participants.
- One participant did not like the standard-buttons and wanted to be able to remove them as well.

Bugs:
Some bugs were discovered during testing, these were:

- The accept of position popup was shown when moving an existing button, however, the user liked this feature.

- When editing a button's text , the old text would reappear in the edit-popup when pressing edit text again.
- In video mode, one was often required to double tap the play video for the video to play.
- The enter button did not at first work for entering comments, this was fixed halfway through the testing process.

Opinions:
All of the participants were very positive to the prototype and were able to see a use for it in their work. They also felt that had they been able to form the template from scratch themselves and for a test they knew properly, they would have been even better at using it. Some also felt that it was a tool one had to get used to using, especially to learn how, when and for what to use it. One expressed the risk of gathering too much data and data which were not needed. One of them mentioned the drawback of not being able to enter more complete comments during testing.

All participants liked the versatility of the prototype and that they themselves could determine the tags which existed as well as their placements. They thought that it made it possible to group tags together and to create a flow.

4.3.3 Code development

Version 1:

When starting the first version of the prototype a new projects was started and the introduction to C# was interesting and productive. During the developments we have encountered problems with storing the interaction data. Familiar with Java's ArrayList it took us some time to get comfortable with the List selection available for C# projects. There were also some difficulties regarding the API as different ones are sometimes used depending on whether one is developing for WPF or not, and both use the same names but have different properties and methods. One example is the API for the Button class see references [28] and [29].

After assistance from Cenito to setup a subversion project for easy update and storage of the code. It took some time to get to know the methods to upload and update the available code. This resulted in several merge conflicts where the different class versions wasn't up to date. To ease the development and reduce the conflicts we added the classes that we were programming in to a list to get a view of what was going on and where we were working at the moment. When we felt comfortable with the project and we knew what to be implemented the errors got fewer.

Early in the projects we had the vision that we wanted to present the button interactions made during the test session. It started with a outprint presenting that a button click were made which later got saved in a temporary list. When the interactions later got saved according to which task and session it took part of the problem we later had to face was the one where the information needed to be retrieved in the statistic class.
To get the information needed without duplication of classes and lists turned out to be harder than it looked but with some time put into it we later received the desired result.

There were also some issues regarding how one could use icons and text in buttons, especially when using the Windows Symbol GUI, as this font is only able to produce the picture when using XAML and not during runtime through C#. This caused us to need to find a good image editor where one could edit images background and make them transparent, as well as learning to use the program. The program used for this was GIMP 2.

Version 2:

After the half-time testing was performed and the results analysed, the results was used to resolve bugs and to add new functions. After the bugs had been resolved we focused on the updated backlog seen in section 3.3.2, version 2. The separation of task-timer and session-timer gave us the opportunity to optomise the code behind, as well as making it easier for the user to know which timer did what. The timers had earlier consisted of separate entities, they were now created in one entity which contained the others,

a popup. This meant that it was easier to separate what entities belonged to what timer; as well as later on enabling the change of placement of the timers by the user. This inspired us to reuse the popup-optimisation for the add/edit panel; however, this caused some issues and unexpected behavior which had to be resolved. In the end, the quick optimisation took more time than anticipated but enabled an easier adjustment of the interface.

More feedback was added by adding a log to the session-timer, which logs most actions performed during a testing session. The exception is the creation of a new tag. A bug was also resolved so that the color change of tapped buttons could last longer. This would make it easier for the users to know that their tapped tag or action had been registered.
More time than expected was also consumed when enabling the saving and loading of templates, as every change in the code needed to be worked around and be performed in steps. This was due to the need to enable the loading of properties after they had been saved. The result was that one first had to save a new version of a template and then shut down the simulator, enable the code in the load function and then run the simulator again.

Another addition was the function of making the moved tags and objects to be placed in straight lines. Earlier they could be placed very randomly and it was difficult to place them in lines and pairs. By adjusting the placement of objects in the code straight lines were much easier to achieve and the user would not be annoyed by tags that were slightly off from other tags placements.

Version 2.1:

As this was done after/during testing, and in the end of the project, there was not much time left to spend on further development. A final adjustment was that the enter-button could be used to confirm input, such as comments and changing the name of the button. This was needed by request from users in our pilot testing, as well as confirmed during testing. It is much easier to use the enter-button then to use the intended accept-button.

Another quick final adjustment was to make the comment-buttons inputpanel adjust its position according the placement of the comment-button. Without this, the inputpanel would stay in the same place even though the comment-button which enabled it was moved. This was decided to be a flaw which could confuse the user.

Chapter 5

Walkthrough of prototype version 1

This chapter contains a walkthrough of version 1 of the prototype, this will hopefully give a better understanding of how the prototype works and for the terms used in the report.

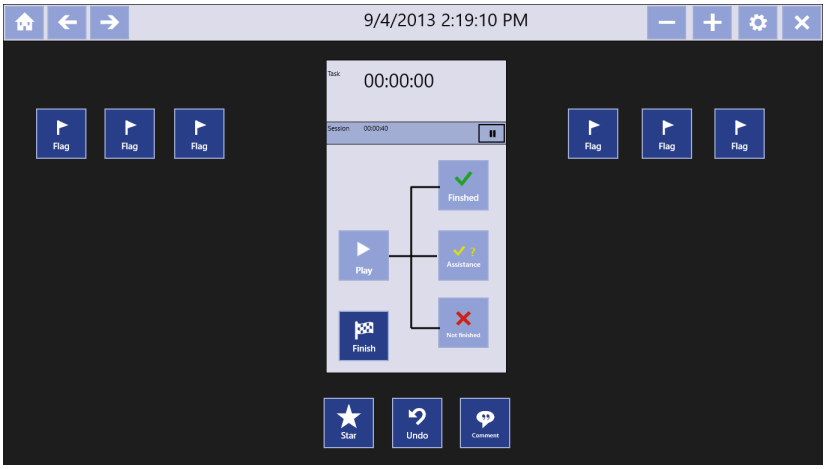

Figure 5.1: The observer template for two-hand testing.

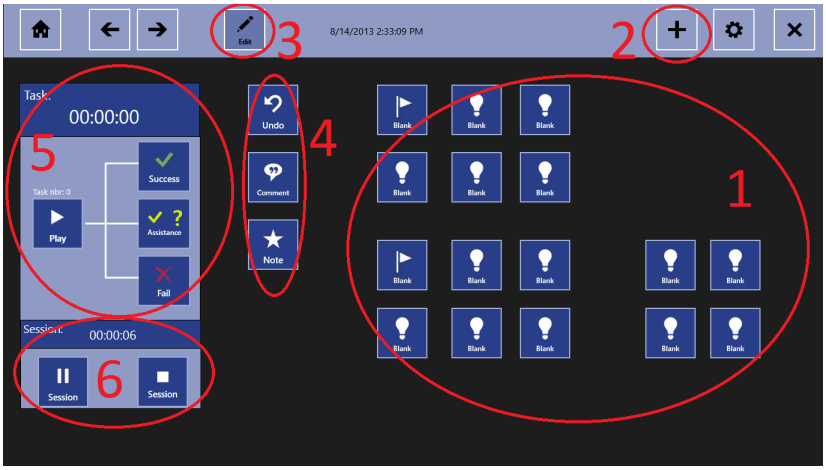

Figure 5.2: The observer template for one-hand testing. 1: Tags for marking an event. 2: The "add"-button, enabled the addage of new tags. 3: Enables the editing of a test template, cancels the session. 4: Default tags 5: The timer-panel for a task. 6: The timer-panel for the test session.

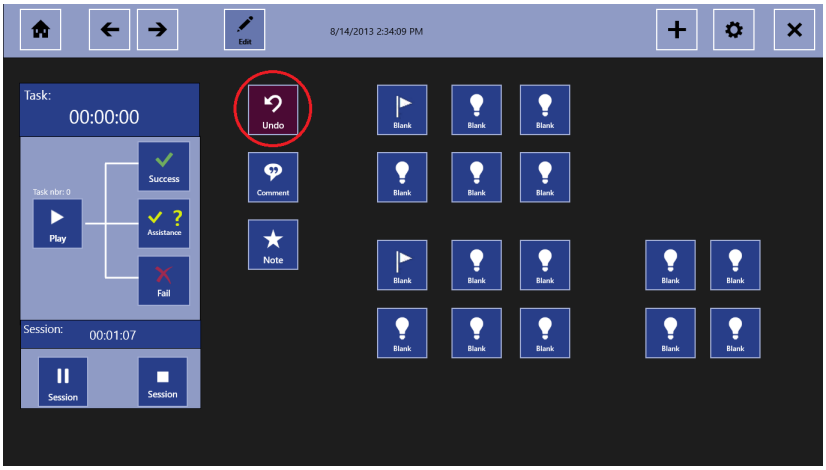

Figure 5.3: A tag has been pressed and has turned purple, see the red-circled button.

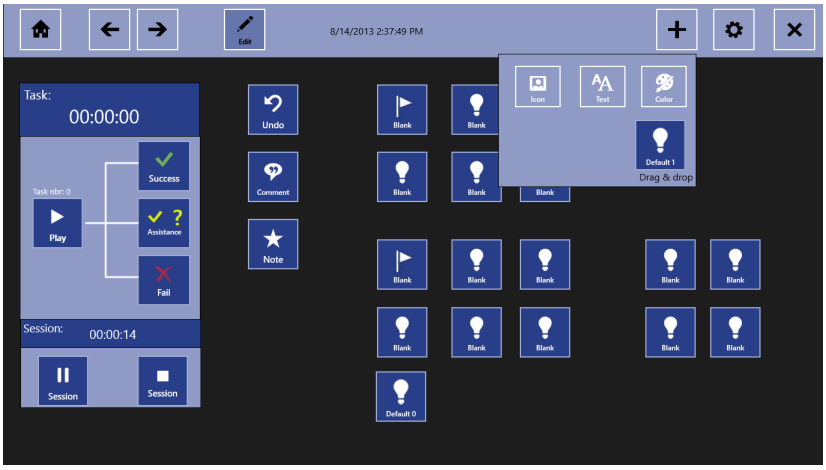

Figure 5.4: When one wants to add a tag one pressed the "+"-button and the add-panel appears. The default tag is shown, with a lightbulb as an icon and the numbered text "Default 1". The number will increase with each defaultbutton used. When finished editing the new tag one presses the new tag and performs a drag-and-drop action and places the tag in its intended position.

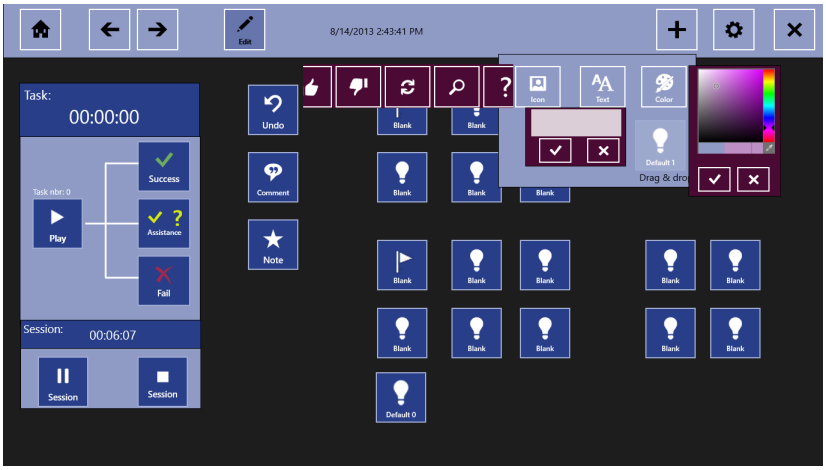

Figure 5.5: The available options for editing a button is shown.

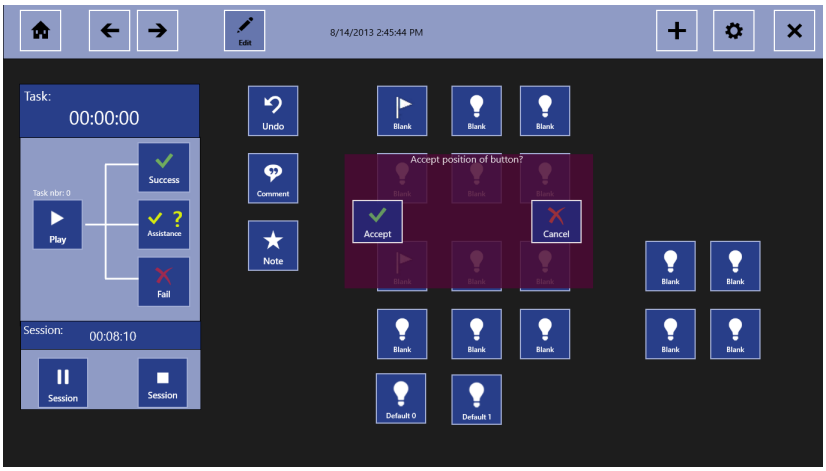

Figure 5.6: The popup which confirms the position of the new tag, if one presses "Accept" one would now not be able to move the tag any longer. If "Cancel" is pressed one can continue to change the new tags position.

Figure 5.7: When one is finished with a test session one presses the stop-icon with the text "Session". The statistic mode then appears. 1: Shows the data gathered during the test. 2: Shows how much time of the session each task required. 3: A combobox for filtering the information in 1 by task.

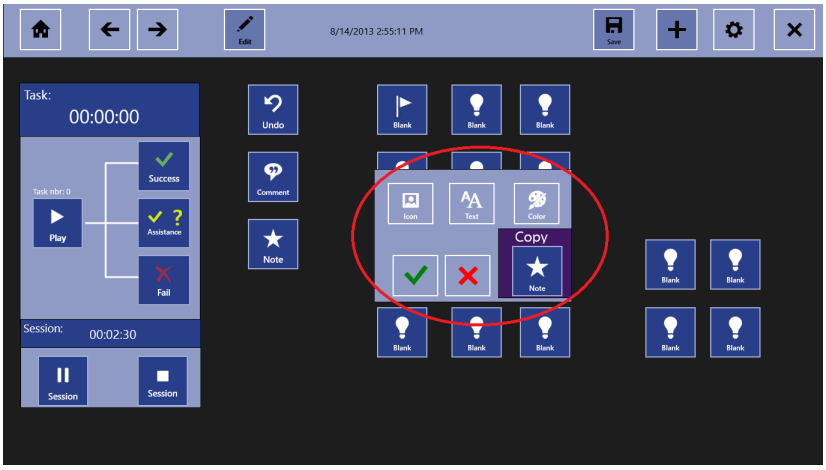

Figure 5.8: In edit-mode, after having pressed button number 2 in Figure5.2. The panel for adding a button has been moved to the center of the screen. When editing an existing tag a copy of the button is shown in the edit-panel on which the changes are shown before confirmation of acceptance of changes. While editing an existing tag one does not perform a drag-and-drop action but either accepts or declines the changes by pressing the corresponding buttons.

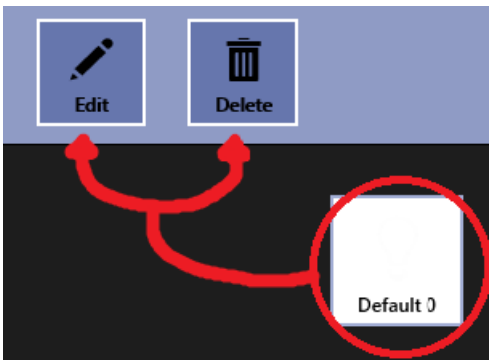

Figure 5.9: When in edit-mode one is able to edit and remove existing tags, this is done by pressing a button and performing a drag-and-drop action and releasing them on the buttons which have appeared in the menu. Position may also be changed by this action as long as the drop is performed below the menu.

Chapter 6

Walkthrough of prototype version 2

Version 2 has mostly the same interactions as version 1, most changes has been done to the interface and not the interaction. For an introduction to version 2 it is recommended to read the walkthrough for version 1, in Chapter 5 first.

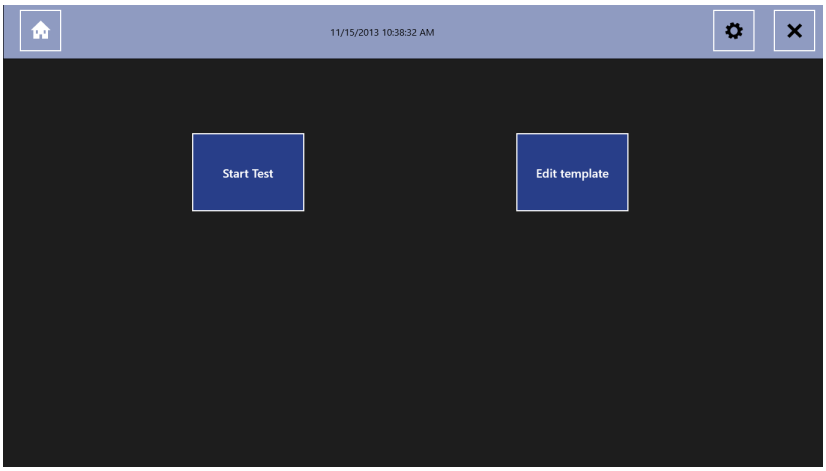

Figure 6.1: The start menu of the prototype, one can start a test and one can edit a template prior to testing.

Figure 6.2: The observer template of the prototype, 1: Shows the new comment buttons, the one above lets the user enter a comment and the one below tags the event of a comment. 2: The new tasktimer-panel, while in edit-mode this is moveable. 3: The new seesiontimer-panel with a log which shows all actions during a session, this is also moveable in edit-mode.

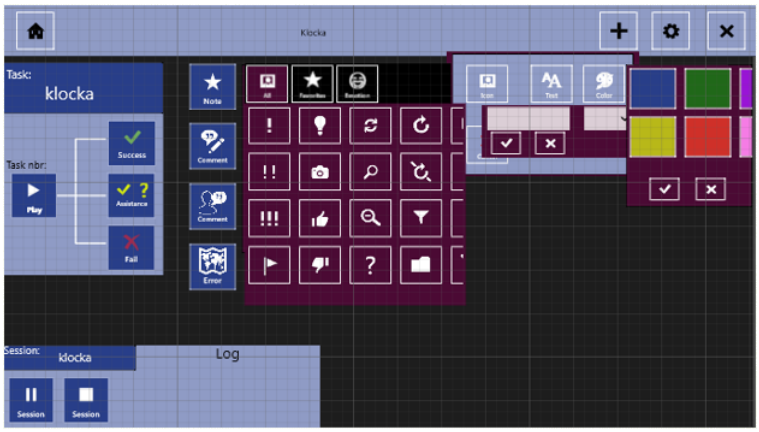

Figure 6.3: The updated interface of the add/edit panel. One is now able to view more icons at the same time, as well as filter the shown icons. One can edit the size of the text as well as change the background of a tag. Both the icon-panel and the color-panel are scrollable. When running the prototype it is not possible to have all panels open at the same time.

Figure 6.4: The updated interface of the popup for accepting the position of a new tag. One is now able to cancel the creation of a new button.

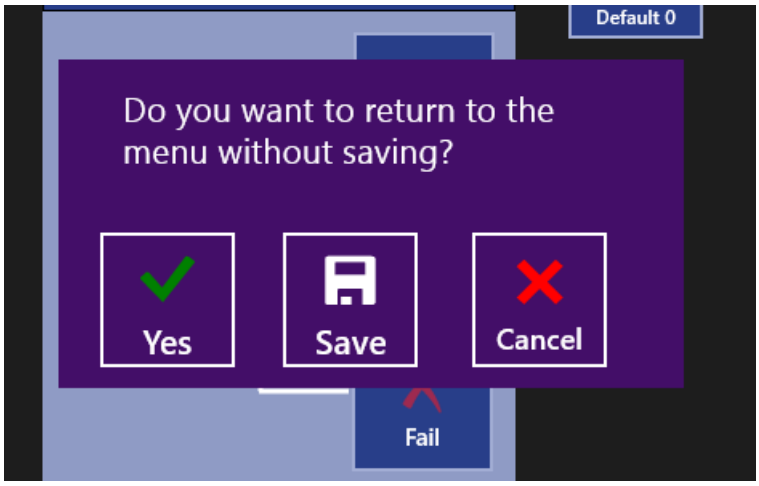

Figure 6.5: While editing and one presses the home-button in the menu, this popup appears. It enables one to regret the action, to save the template and to exit without saving. This does not appear if one has previously saved without further changes.

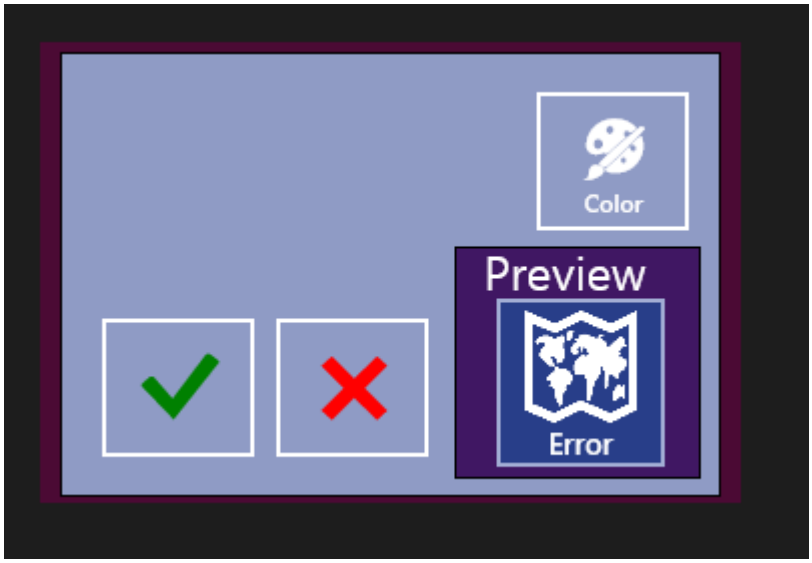

Figure 6.6: The default buttons have their own interaction in edit-mode. It is only possible to edit them and then only their color. This figure also shows an example of a preview of a changed tag in edit-mode.

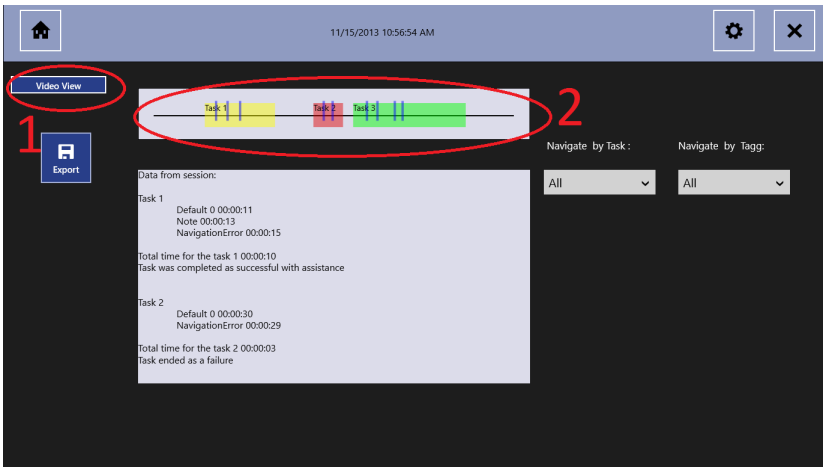

Figure 6.7: The updated interface of the statistics. 1: Switches to video mode. 2: A timeline which enables jumping between tasks, it also shows markings of where in the session tags were tapped.

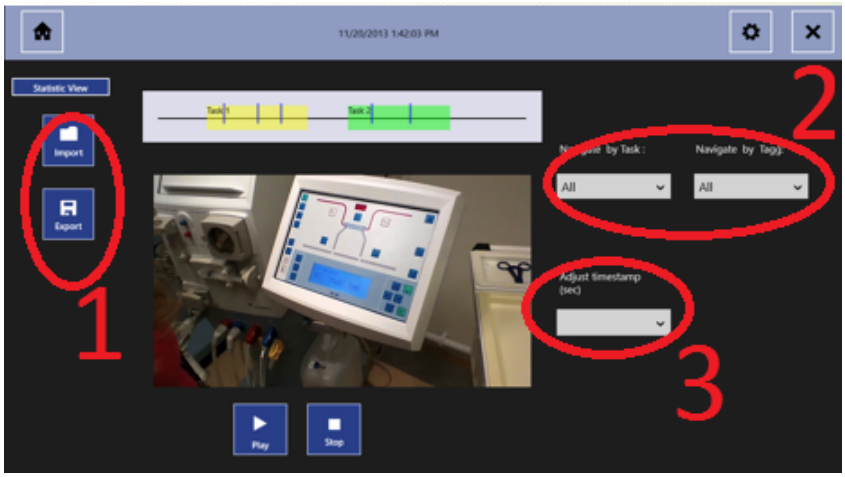

Figure 6.8: The updated interface for viewing a video. 1: Is is possible to import a video and also saving the data by pressing "export". 2: The comboboxes allows the user to jump in the movie by both task and tag. 3: The Offset combobox allows one to adjust the offset of the video so one can jump to the correct moment in the video.

Chapter 7

Discussion and Conclusion

7.1 Design decisions

During the development we faced different areas that required a closer look and a clearer explanation. Reflections sometimes arose during testing and during discussions with our supervisors.

Feedback from the tablet using sound or haptics feedback e.g vibrations have been excluded from the prototype. This to ease the interactions and avoid distractions during the test sessions. As a test person repeatedly hears sound or vibrations according to an interaction, positive or not, may lead to a negative test experience and/or result. It may also interfere with the feedback of the tested interface. The negative expericent or result might be due to the test person avoiding certain types of interactions, by taking longer due to increased thoughtfulness or by becoming self-conscious. A software which is used for observation purposes has higher requirements on being discreet than an ordinary interface.

When in edit mode and a user wants to change the appearance of an existing button the user has to perform a drag and drop move and place the button on top of the edit button in the top menu, see Figure 7.1. This movement caused the user some confusion and

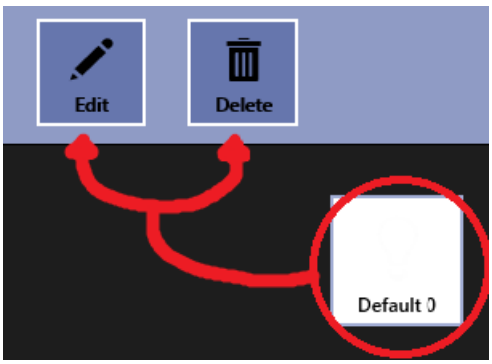

Figure 7.1: Shows the interaction of drag and drop on the edit and remove options. The whitelighted button is the button which is pressed and dragged.

difficulties during the interaction. Windows phones and tablets uses a hold-press interaction to change the settings of an application in the desktop mode see reference [24]. Thoughts were raised that either the interaction with the drag-drop interaction and/or the edit button were not distinct enough. For the next version we have to look further into how the interaction can be eased.

7.2 Fulfillment of aims

The project had four main goals, mentioned in section 1.2, which we believe has been mostly fulfilled. From our final test we gathered information that confirmed that the prototype could simplify analysis of information gathered during usability testing. It was also confirmed that the prototype was very easy to use while gathering information, which in the long run simplified the gathering of data. Another confirmation was the fact that it really worked to use a tablet for information gathering, all of the testers felt comfortable using the tablet and liked using it.

Both the halfway test and the final test confirmed that the prototype can be a useful tool for a large user group, the main point is to be able to import and use icons adjusted for specific tests.

We have however not fulfilled goal number two completely as the PC part has not been developed as of yet, this is due to lack of time and the focus being placed completely on the tablet.

However, the prototype is not finished and needs to be improved and finalised. When the product is finished it will even better fulfill our goals than it does at the moment. The main point is that all users which have tested the prototype likes it and can see themselves using it in the future.

7.3 Further development

The prototype needs further development before it can be called a proper product. Below is different parts we believe should be improved or new functions/aspects which should be added. Overall we also believe that one should be able to save data to excel, as all of the participants wanted to be able to this in a final product.

Further develop the tablet:

- Upgrade graphics and existing functions:
  - The timelines in post-test mode should be more interactive and be possible to filter more according to tags. One idea is to show icons above each tag, however, this should only be possible while in one task or when zoomed in.
  - When in video mode, the video should have a corresponding timeline, which adjusts to the length of the movie and shows where one is. This might be combined with the timeline representing the test session.
  - Adjust the code so the input-panel for one of the comment-tags does not end up outside of the screen when the comment-tag is placed to the far right.
  - Add another way to enable editing by either making a popup show up after every move of a tag or by two small button appearing on either side. The first option would contain the options to not approve the new position, edit and remove button, the second options button's buttons would be remove and edit.

- Enable an undo-function were one regrets the last tapped button, should be positioned in the menu as our tests have shown us that placement with tags confuses the user of its function. Many believed it registered the users action and others were insecure what or how much it undid. It should also be complemented by a popup which checks whether one really wants to undo the latest tapped action.
- Enable a help-function or mode, which can explain the function of certain buttons. Buttons which especially needs this are: the comment-buttons and the future undo-button. Make it possible for the user to create their own tags with attached comment-functions. The standard-buttons should be removable, but always be present when starting a new testing template. They should also be reproducible.
- During post-test, both statisticview and video mode, the content of a comment should be viewable as well as editable. The user should also be able to add new comments.

Implementing a PC part of the product: This should complement the tablet, especially regarding functions and actions which are not very easily performed on the tablet. Functions should include:

- One should be able to review a test session and edit/remove existing tagged events as well as add new ones as needed, this includes adding new tags.
- Video viewing of multiple test participants at the same time, for example being able to compare Task 1 done by every participant or specific ones.
- The function to combine multiple test sessions for the same testing project to simplify comparison.
    * Video viewing of multiple test participants at the same time, for example being able to compare Task 1 done by every participant or specific ones.
    * Statistics of all participants.
- Be able to combine multiple tablets test session records into one.
- Preparation of test:

* The function to add new icons and colours to the edit mode, one can for example add an “add button” to the menu for filtration of icons.
* Be able to change the colour of the icons.
* Being able to prepare layout of test, such as how many tasks, naming of tasks and how much time the approved limit is. One should also be able to create a checklist for steps taken during a task.
* There was a wish for the option of having different layouts and different tags depending on what task was active. This could be done by using tabs as in a web browser and/or by enabling a swipe-movement for change between templates.
* Otherwise have the same functions as when using the tablet.

7.4 Conclusion

The main conclusion one can draw is that there is definitely a need for a product which can increase the effectivity of usability testing. This point has been mentioned during all phases of the project, from the pre-study to the final testing. Another conclusion is that our existing prototype is able to help with increasing the effectivity and is a tool which at least our test participants would like to use in their work. The prototype however needs additional work before it can be used in full-scale and still lacks some functions which the users would like to have, but as it is a prototype this is normal.

The result from the final testing was what we had expected due to us performing user-centered development. This made it possible to almost always know what the possible users wanted and to rectify deficiencies and create something that they wanted instead of only what we thought they wanted. A concern might be that some of the results might have been colored by our own ideas of how things should be, but as problems appeared during testing we tried to fix them.

A possible issue is that we could have performed more tests during the final testing phase to really confirm the validity of our claims. We felt that this was not needed due to the half-time testing which had already confirmed some and due to lack of time to divest on further testing in other environments. Another reason was that we considered that further testing would probably not be worth the time we spent on it and decided to make the best of the testing we did perform. The scope of the project was limited by our focus on using a tablet as a solution to the problem of time consuming testing, as there might have been other solutions which would work just as well or even better.

The main goals of the project were:

1. Simplify the gathering of data during usability tests.
2. Simplify the analysis of information gathered from usability tests.
3. A useful tool designed for a large user group.
4. Create a compatible system for use on both a tablet and PC.

We believe we have fully fullfilled number 1 to 3, however number 4 was not prioritised and was therefore not implemented.

In conclusion, we did receive the results we expected with some smaller additions we had not thought of earlier. One of the results we were expecting were that it would be possible to use a tablet during usability testing. Both for the gathering of data as well as for the analysation of said data. Another was that the users would like the product and want to use it for real later. An addition we were not expecting was when participants, and later our supervisor at Cenito, expressed the idea for using the tablet for taking snapshots or short videos. We had mainly focused on it being inconvenient to use the tablet as a camera for whole testing sessions and forgotten that one might want to record short parts or moments. Finally, it was also surprising to see how large the result of one idea could become.

We also managed to create more functions and implement more parts of the prototype than we had expected from the beginning, which made it even easier to test the concept of the final product.

At first we only thought we would manage to implement the gathering of data and perhaps parts of the analysation of the data. In the end we managed to implement the gathering, the analysation in both data and video and the editing of a test template. To be able to save and/or load templates or data gathered was something we almost did not dare to hope we would have time to implement. In the end we managed to create a more complete prototype than we had expected, but we wished we would have had more time and been able to implemen even more functions and parts.

References

[1] *"W3.org"*, [electronic], Available HTTP: `http://www.w3.org/2002/Talks/0104-usabilityprocess/slide3-0.html`, [2013-12-05]

[2] *Designing the user interface*, Ben Shnedierman, Catherine Plaisant, Pearson Education, 2005

[3] *Handbook of Usability Testing*, Jeffrey Rubin, Dana Chisnell, Wiley Publishing Inc, 2008.

[4] *Designing for usability: key principles and what designers think.* Gould, J. D., Lewis, C. Communications of the ACM, 1985, 28, pages 300-311.

[5] *The Design of Everyday Things*, Norman Donald A., Basic Books, 1988.

[6] *Summary of Don Norman's Design Principles*, Norman Herr, Ph.D, [electronic], Available HTTP: `http://www.csun.edu/science/courses/671/bibliography/preece.html` [2013-11-15]

[7] *Design Principles*, David Gelb, [electronic], Available HTTP: `http://www.slideshare.net/gelvan/design-principles`

[8] *Undo and Erase Events as Indicators of Usability Problems*, David Akers,Matthew Simpson, Robin Jeffries and Terry Winograd, [electronic], Available HTTP: `http://hci.stanford.edu/publications/paper.php\?id=67` [2013-11-15]

[9] *A Methodology and Framework to Simplify Usability Analysis of Mobile Applications, part 3*,Balagtas-Fernandez. F, Hussmann. H , [electronic], Available

HTTP: http://ieeexplore.ieee.org.ludwig.lub.lu.se/xpl/articleDetails.jsp?arnumber=5431743

[10] *Technical review: Current issues of Usability testing*, Alshamari, Majed, Mayhew, Pam, [electronic], Available HTTP: http://ehis.ebscohost.com.ludwig.lub.lu.se/eds/detail?sid=fa511e86-f965-4e19-a2d9-d318cb130b37\%40sessionmgr111&vid=1&hid=102&bdata=JnNpdGU9ZWRzLWxpdmUmc2NvcGU9c2l0ZQ\%3d\%3d#db=a9h&AN=45526152 , [2013-11-15]

[11] *Usability measurement and metrics: A consolidated model*, Ahmed Seffah, Mohammad Donyaee, Rex B. Kline, Harkirat K. Padda, [electronic], Available HTTP: http://www-psychology.concordia.ca/fac/kline/library/sdkh06.pdf [2013-11-15]

[12] *How to Measure and to Quantify Usability Attributes of Man-Machine Interfaces*, M. Rauterberg, [electronic], Available HTTP: http://www.idemployee.id.tue.nl/g.w.m.rauterberg/publications/ROMAN961paper.pdf [2013-11-15]

[13] *10 Essential Usability Metrics*, Jeff Sauro, [electronic], Available HTTP: http://www.measuringusability.com/blog/essential-metrics.php [2013-11-15]

[14] *Performance-based Usability testing: Metrics That Have the Greatest Impact for Improving a System's Usability*, Robert W. Bailey, Cari A. Wolfson, Janice Nall, Sanjay Koyani, [electronic], Available HTTP: http://link.springer.com.ludwig.lub.lu.se/chapter/10.1007\%2F978-3-642-02806-9_1 , [2013-11-15]

[15] *QSR International*, [electronic], 2013, Available HTTP: http://www.qsrinternational.com/default.aspx, [2013-06-30]

[16] *QSR International, NVIVO, "NVivo 10 feature list"*, [electronic], 2013, Accessible HTTP: http://download.qsrinternational.com/Resource/NVivo10/nvivo10-feature-list.pdf, [2013-06-30]

[17] *Noldus Information Technology*, [electronic], 2013, Available HTTP: http://download.qsrinternational.com/Resource/NVivo10/nvivo10-overview.pdf [2013-06-30]

[18] ,Noldus, [electronic], 2013, Available HTTP: http://www.noldus.com/human-behavior-research/products/the-observer-xt [2013-06-30]

[19] *Morae*, [electronic], 2013, Available HTTP: http://www.techsmith.com/morae.html [2013-06-30]

[20] *"The Usability Testing Environment (UTE)"*, Mindd, Mind Design Systems, [electronic], Accessible HTTP: http://www.mindd.com/Content.aspx?pid=UTEStandard [2013-06-30]

[21] *"Crowdsourcing for Usability Testing"*, Di Liu, Matthew Lease, Rebecca Kuipers, and Randolph Bias, [electronic], Available HTTP: http://arxiv.org/pdf/1203.1468v2.pdf [2013-06-30]

[22] , Userlytics, [electronic], 2013, Available HTTP: http://www.userlytics.com/why-userlytics, [2013-06-30]

[23] *"10 Usability Testing Tools for Startups"*, Monji Dolon, [electronic], Available HTTP: http://devgrow.com/10-usability-testing-tools-for-startups/, [2013-06-30]

[24] *Touch interactions*, msdn Microsoft, [electronic], Available HTTP: http://msdn.microsoft.com/en-us/library/windows/apps/hh465415.aspx, [2013-11-06]

[25] *Guidelines for Segoe UI Symbol icons*, Msdn Microsoft, [electronic], Available HTTP: http://msdn.microsoft.com/en-us/library/windows/apps/jj841126.aspx [2013-11-18]

[26] *Creating app bar buttons in WinRT*, Nogginbox, [electronic], Available HTTP: http://www.nogginbox.co.uk/blog/app-bar-buttons-in-winrt [2013-11-18]

[27] *"Icons DB free custom icons"*, [electronic], Available HTTP: http://www.iconsdb.com/black-icons/, [2013-10-28]

[28] *"Windows Dev Centre- Button"*, [electronic], Available HTTP: http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.button.aspx [2013-11-18]

[29] *"Microsoft Developer Network -Button"*, [electronic], Available HTTP: http://msdn.microsoft.com/en-us/library/system.windows.controls.button(v=vs.110).aspx [2013-11-18]

[30] *An empirical approach for evaluating usability and model driven tools*, Nelly Condori-Fernández,Jose Ignacio Panach, Arthur Iwan Baars, Tanja Vos, Óscar Pastor, [electronic], Available HTTP: `http://www.sciencedirect.com.ludwig.lub.lu.se/science/article/pii/S0167642312001451` [2013-11-15]

Chapter 8

Attachments / Bilagor:

8.1 Interview questions 1

Test

- Hur jobbar du när du gör tester/observationer?
- Vad är det som tar tid vid användbarhetstester/observationer?
  - förberedelser?
  - utförande?
  - efterarbete?
- Vilka är de vanligaste problemen som uppstår vid testning?
  - Fel på hjälpmedel?
  - Tidsbrist?

Dokumentation

- Vad för data brukar du samla in vid testning?
  - Vilka hjälpmedel används?
- Hur mycket tid brukar du lägga på att gå igenom material efter testning, (anteckningar, insamlad data och inspelat material)
- Vad för data skulle du vilja kunna samla in, men som du inte hinner/kan idag?
- Vad för data skulle du vilja samlades in automatiskt?

Platta

- Hur skulle du se på att använda en platta för att samla data vid tester?
- Önskar ni en möjlighet att snabbt kunna återkomma till en viss sekvens? En taggning?
  - Vid användande av taggning i en film som markering av händelse, vilka taggar skulle du vilja ha?
    - Rätt/Fel användning?
    - Användarfel?
    - Demonstration av prototyp
- Finns det önskemål om att smidigt kunna föra information mellan olika enheter?
- Vill ni kunna gruppera dokument/testfiler?
- Vill ni snabbt få information om hur väl ett test genomfördes? Statistik?

8.2 Observation questions

- Vad för typ av data samlade ni in?
- Vad för events förde ni anteckningar om?
- Önskar ni smidigt att kunna effektivisera analysarbetet genom att få snabb statistik över hur väl testet gått?
  - Vad för typ av data skulle ni vilja få isf? antal rätt/fel? etc.
- Vid anteckning, känner ni att ni får den tid som krävs för att få ner informationen som krävs för analys?
  - Skulle ni kunna tänka er att samla in datan med hjälp av en surfplatta och ”taggning” som markerar en händelse?
  - Om taggning vilka funktioner skulle vara passande vid era användartester?

8.3 Data gathered

8.3.1 Performance data

- Comprehensibility (readability) [9]
  - Single usability metric (SUM) combination of metrics for comparison to other products/versions [13]
- Effectiveness [10] [9] [11]
  - task completion [14], [9] [12]
    - binary [10] [13]
  - task completion percentage [30] [11]
  - tasks performed correctly and optimally [30]
  - first click for task (correct/incorrect) [14]
  - accuracy [10]
  - completeness [10]
  - quality of outcome [10]
  - the usage of help [11]
- Efficiency [10] [9] [11] [12]
  - task completeness divided with time [30]
  - task completion time [10] [14] [9] [11] [12] [13] (Schneiderman)
  - input rates [10]
  - effort: [9] [11]
    - mental effort [10]
    - task level satisfaction, how difficult task was experienced [13]
    - test level satisfaction, how difficult whole test was experienced [13]
    - expectancy of task's difficulty [13]
    - number of menu levels used for each task [9]
    - number clicks required for each task [9]
    - time used to search for button for a specific function. [9]
  - learning time [10]
  - use frequency [10]

    - undo/erase events [8]
    - number page view required for each task [14] [13]
    - time spent dealing with program errors [11] [13]
    - error free task time [12]
- Error [10] [9] [13]
    - probability of specific error [13]
    - situation when error encountered [13]
    - rate of errors [10] [12] (Schneiderman)
    - number of errors [9] [12]:
        - detour steps [9]
        - deviating button clicks [9]
        - wrong answers [9]
    - slips [12]
    - recognition errors [12]
- Learnability [10] [9] [11]
    - percentage of correct answers to predetermined questions [9]
    - time to learn (Schneiderman)
- Learning performance [10] [9]
- Memorability [10] [9]
    - retention over time (Schneiderman p 16)
- simplicity (complexity) [9]

8.3.2 Preference data

- Satisfaction[10] [9] [11]
    - questionnaires [14] [10] [11]
    - comments [14]
        - like/dislike [9]
    - preferences [10]
    - satisfaction with interface [10]
    - feelings throughout tasks/scenarios [10] (emotion card tool for every task)
    - preferences [12]
    - subjective satisfaction [12] (Schneiderman)

8.4 Test documents

8.4.1 Task list, test 1

Scenario

1. Du vill starta en test-session, välj mall “Observer 1”.
2. När du känner dig redo, välj att starta ett nytt scenario.
3. När scenariot är startat så vill du registrera några händelser som sker i rummet.
   (a) Bokmärk en händelse
   (b) Markera en kommentar som du vill kolla tillbaka på senare.
4. Avsluta scenariot som lyckat
5. Starta ett nytt Scenario
   (a) Bokmärk en händelse
   (b) Testpersonen ångrar ett knapptryck och markera händelsen på plattan.
6. Du upptäcker att du saknar en knapp för att markera en händelse. Pausa sessionen och lägg till en ny knapp med funktionen:
   (a) Namn: Avbrott
   (b) Symbol: Lampa
   (c) Färg: Lila
7. Testpersonen, fick mycket hjälp under detta scenario, avsluta det som lyckat fast med assistans.
8. Avsluta sessionen.
9. Gå tillbaka till huvudmenyn och välj nu “Observer 2”
10. När du känner dig redo, välj att starta ett nytt scenario.
11. När scenariot är startat så vill du registrera några händelser som sker i rummet.
    (a) Bokmärk en händelse
    (b) Markera en kommentar som du vill kolla tillbaka på senare.
12. Avsluta scenariot som lyckat
13. Starta ett nytt Scenario

(a) Bokmärk en händelse
(b) Testpersonen ångrar ett knapptryck och markera händelsen på plattan.

14. Testpersonen, fick mycket hjälp under detta scenario, avsluta det som lyckat fast med assistans.
15. Avsluta sessionen
16. Du kopplas nu vidare till statistik sidan där du får information om de händelser som samlats in under sessionen.
    (a) Navigera mellan de olika scenariorna och få information om vad som har skett under sessionen.
17. Känner du dig nöjd? Gå tillbaka till huvudmenyn
18. Du ska nu testa att föreditera ett gränssnitt, gå in i Observer 1 och tryck på editknappen.
19. Du kommer på att du fattas en funktion för att spara in när en bugg sker och vill lägga till denna. Skapa en ny tagg med funktionen:
    (a) Namn: Bugg
    (b) Symbol: Frågetecken
20. Du upptäcker att du vill byta färg på knappen, ändra den till lila.
21. Du ångrar att du lagt till en funktion för taggning och väljer därför att ta bort taggen.

Debrief

- Hur kändes interaktionen?
- Kändes knapparna lagom stora att trycka på?
  * Var texten tillräckligt stor?
  * Var bilderna tillräckligt stora?
- Hur kändes "Session/Task" -delen,
  * Var det tydlig skillnad på session och task
  * Länkades knapparna samman på ett smidigt sätt?
  * Kunde man skilja de olika Task-avsluten åt på ett enkelt sätt.
- Hur upplevdes editerings delen?

    - Tydligt hur man namngav/valde ikon?
- Statistik, kändes informationen tillräcklig?
    - Vad kan förbättras?
    - Vad skulle du vilja veta mer om..

8.4.2 Testplan, final testing

Syfte övergripande mål

Testerna kommer att ligga i grund för pågående Examensarbete med förhoppning om att verktyget var syfte är att effektivisera avändbarhetstester och observationer, får utövas i sin naturliga miljö och bevisa effektiviteten i arbetsområdet.

Problemformuleringar

- En stor del den insamlade test-datan från dagens användbarhetstester består utav anteckningar. Något som tar stor fokus under testerna och sedan tar lång tid i efterarbetet för att både "avkoda", renskriva och analysera. Kan detta effektiviseras med hjälp av en testprotokoll via surfplatta?
- Kommer större vikt att läggas på det inspelade videomaterialet om det smidigt kan återkopplas mot de sparade taggarna.

Urval av försökspersoner Vi tänkte använda oss av 4-5 personer för utförandet av testet, inga krav på tidigare erfarenheter av liknande program krävs dock bör försökspersonerna har erfarenhet av att tidigare lett eller deltagit i användbarhets tester/utvärderingar.

Upplägg för genomförande Genomförandet kommer att ske i Gambros testlokal. Försökspersonerna kommer att få genomgå en testomgång där de både kommer att få förprogrammera ett testprotokoll som sedan kommer att få användas på plats under ett i scensatt test för att med den insamlade datas hjälp både få se sammanställlningsverktyget samt navigera med dess hjälp i videoverktyget.

Lista över testuppgifter

- Förberedelse utav testmall

- Genomförande av “användbarhetstest” med hjälp av den skapade mallen
- Med hjälp av den insamlade datan ta del av möjligheterna med taggarna och video informationen som samlas in.

Testmiljö och testutrustning
Utrutstningen som kommer att användas utav deltagarna är filmkamera och en Windows RT tablet.

Testledararens roll
-

Data som ska samlas in

- Filmdata
  * Film över hur testet ser ut genom försökspersonens ögon
  * Film över hur testet ser ut genom testledarens ögon
- Information om användarmönster.
  * Hur skapas mallarna?
  * Hur är känslan av att använda surfplattan under test?
  * Hur är känskan av att använda surfplattan för genomgång av data efter test?

8.4.3 Task list, test 2

Del 1 För att få en introduktion till verktyget så ska testpersonen öppna mallen Gambro.
Visa de olika delarna och förklara skillnaderna:

- Task
- Session

Visa testpersonerna knapparna och låt försökpersonen lägga till knappar för känslor, efter addering låta hen kolla på mallen och eventuellt gruppera döpa om knappar efter eget förslag. När nöjd spara som gambro$_m$*ittnamn*.

Del 2 Observation kommer att ske ansvara för insamlingen av data. Observation kommer att ske på försöksp för att se hur den navigerar och trycker.

- två deluppgifter
  - förprogrammera inställningar
  - koppla på slangar
- När testet är klart så ska den trycka på stop.

Värden som ska programmeras in för behandling är:

Behandlingstid: 2 timmar
Isol UF tid: 15min
volym: 0,5 l
Uf 1.5 l
Heparin Bolus: 5 ml
kontinuerlig 3.5 ml

Del 3 Vid stopp låta den navigera på statistik sidan och själv lära känna verktyget, och se dess funktioner. För att sedan gå över till video: Hämta video Test_gambro, lära känna och navigera mellan de olika taggarna här med för att sedan testa på att sälla in offset på två utav taggarna. När klar spara testet som gambro_mittnamn.

Avslut och debriefing

- Överlag hur uppfattade du testet?
- Hur är känslan av att använda surfplattan under test?
  - Hur kändes navigeringen?
  - Hur kändes det att samla in informationen?
- Hur är känslan av att använda surfplattan för förberedande utav test?
  - Hur kändes navigeringen?
  - Någon funktion du saknar?

- Hur är känslan av att använda surfplattan för genomgång av data efter test?
  - Hur kändes navigeringen?
  - Någon funktion du saknar?
- Är verktyget ett komplement som du skulle kunna tänka dig att använda i framtida tester?
  - Varför/varför inte?

8.5 Source of icons

Figure 8.1: Icons used from IconDB.com, [27]

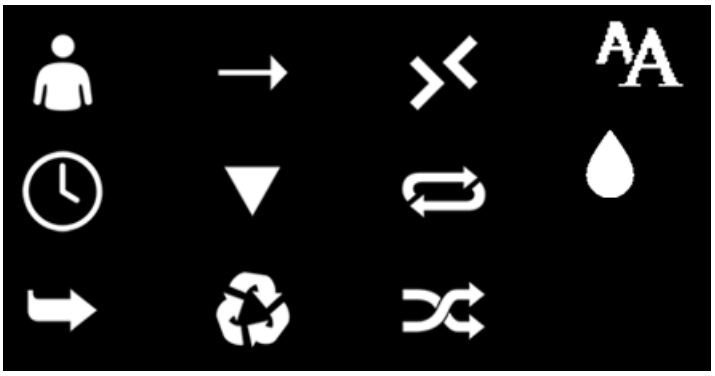

Figure 8.2: Icons used from IconDB.com, part 2, [27]

Figure 8.3: Icons which were edited or created from scratch